The Dissertation Committee for Sehyun Yun
certifies that this is the approved version of the following dissertation:

Sequential Monte Carlo Filtering with Gaussian Mixture Models for Highly Nonlinear Systems

Committee:

Renato Zanetti, Supervisor

Maruthi R. Akella

Moriba K. Jah

Brandon A. Jones

Christopher N. D'Souza

Sequential Monte Carlo Filtering with Gaussian Mixture Models for Highly Nonlinear Systems

by

Sehyun Yun

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2021

To my wife, Anzy

Acknowledgments

I would like to express my deep and sincere gratitude to my adviser, Dr. Renato Zanetti, for giving me the opportunity to do my research and providing invaluable guidance and support over the course.

I would like to thank Dr. Maruthi Akella, Dr. Moriba Jah, Dr. Brandon Jones, and Dr. Christopher D'Souza for serving as my committee members. I appreciate your comments and suggestions on my research and contributions on this dissertation.

I would especially like to thank my office mates, Simone Servadio, Rahul Moghe, Siddarth Kaki, Arjun Ram, and Dr. Kirsten Tuggle. Thank you all for making my Ph.D. journey such an enjoyable one.

Thanks to my family for providing me unconditional support and encouragement throughout the course. I would also like to express my deepest appreciation to my wife, Anzy, for her love and support during the journey. You helped me endure all the hardness of a Ph.D. in UT Austin.

This work was sponsored in part by the U.S. Air Force Office of Scientific Research under grant number FA9550-18-1-0351 and DARPA (Defense Advanced Research Projects Agency) under STTR contract number 140D0420C0062.

Sequential Monte Carlo Filtering with Gaussian Mixture Models for Highly Nonlinear Systems

Publication No. _____

Sehyun Yun, Ph.D.
The University of Texas at Austin, 2021

Supervisor: Renato Zanetti

This dissertation presents two different Bayesian approaches for highly nonlinear systems with a theoretical study on combining the benefits of the Gaussian sum filter and particle filter; the posterior particles of a particle filter are drawn from a Gaussian mixture model approximation of the posterior distribution. The first approach introduces the methods which change each and every particle of a particle filter into a Gaussian mixture component, either using the properties of Dirac delta function or using kernel density estimation; the former treats each particle of the prior distribution as a Gaussian component with a collapsed zero covariance matrix and the latter estimates the covariance matrix of a Gaussian component using the kernel density estimation algorithm. The Gaussian sum filter is then used to calculate the posterior distribution. The second approach uses clustering algorithms. These clustering algorithms are used to recover Gaussian mixture model representation of the prior probability density function from the propagated particles. The

expectation-maximization clustering algorithm and modified fuzzy C-means clustering algorithms are applied to this approach. Under the scenarios considered in this study, it is shown through numerical simulations that the proposed algorithms lead to better performances than the existing algorithms such as Gaussian sum filters and particle filters.

Table of Contents

Acknowledgments **v**

Abstract **vi**

List of Tables **xi**

List of Figures **xii**

Chapter 1. Introduction **1**
1.1 Contributions of the Dissertation 7

Chapter 2. Background Material **9**
2.1 Linear Filters for Nonlinear Systems 10
2.1.1 The Extended Kalman Filter 10
2.1.2 The Unscented Kalman Filter 11
2.2 Nonlinear Filters for Nonlinear Systems 13
2.2.1 The Gaussian Sum Filter 14
2.2.2 The Particle Filter 16

Chapter 3. Sequential Monte Carlo Filtering with Gaussian Mixture Sampling **18**
3.1 Comparison with State of the Art 19
3.2 Algorithms Development 22
3.2.1 Algorithm I - Sampling from a GMM Posterior 23
3.2.2 Algorithm II - Importance Sampling from a GMM Posterior . 26
3.2.3 Algorithm III - Estimation with Non-Zero Initial Covariance . 28
3.3 Numerical Results 31
3.3.1 Single Step Example 31
3.3.2 Univariate Nonstationary Growth Model 33

3.3.3 Lorenz96 system 36
3.3.4 The Blind Tricyclist Problem 41
3.4 Chapter's Summary 45

Chapter 4. Kernel-Based Ensemble Gaussian Mixture Filtering for Orbit Determination with Sparse Data 47
4.1 Motivation and Comparison with State of the Art 48
4.2 System Models 54
4.2.1 Dynamics Model 54
4.2.2 Measurement Model 55
4.2.3 Coordinate systems 56
4.3 Estimation Techniques 57
4.3.1 Adaptive Entropy-based Gaussian Mixture Information Synthesis 57
4.3.2 Modified Kernel-Based Ensemble Gaussian Mixture Filtering 61
4.4 Numerical Results 63
4.5 Chapter's Summary 75

Chapter 5. Expectation-Maximization Clustering in Particle Gaussian Mixture Filters for Light-Curve Data Processing 77
5.1 Motivation and Comparison with State of the Art 78
5.2 System Models 83
5.2.1 Dynamics Model 83
5.2.2 Measurement Model 87
5.2.3 Filter States 89
5.3 Nonlinear Estimation Techniques for Highly Nonlinear Systems . . . 90
5.3.1 Particle Filter with Gaussian Mixture Models 90
5.3.2 Truncated Interval Unscented Kalman Filter 93
5.3.3 Truncated Extended Kalman Filter 98
5.4 Numerical Results 101
5.5 Chapter's Summary 115

Chapter 6. Clustering Methods for Particle Filters with Gaussian Mixture Models **118**

6.1 Motivation and Comparison with State of the Art 119

6.2 Particle Filter with Gaussian Mixture Models 121

6.3 Fuzzy C-Means Clustering 123

6.4 Clustering Methods for PFGMM 126

6.4.1 KLFCM with Weighting Exponent 126

6.4.2 KLFCM with Ridge Regularization 129

6.4.3 Analysis of the Proposed Clustering Methods 133

6.5 Numerical Results 136

6.5.1 Single Step Example 136

6.5.2 Vector Nonstationary Growth Model 139

6.6 Chapter's Summary 143

Chapter 7. Conclusions **144**

Bibliography **147**

List of Tables

3.1 Results of Example 1. Number of effective particles 33
3.2 Results of Example 2. RMSE for 200 Monte Carlo simulations . . . 36
3.3 Results of Example 3. RMSE for 100 Monte Carlo simulations . . . 39
3.4 Results of Example 4: RMSE for 100 Monte Carlo Simulations . . . 45

4.1 Three-component splitting library 60
4.2 Monte Carlo averaged RMSE, SNEES, and computation time for 100 simulations 67
4.3 Monte Carlo averaged RMS position errors for all the six cases . . . 71
4.4 Monte Carlo averaged RMSE, SNEES, and computation time for the ten-orbits case 73
4.5 Monte Carlo averaged RMSE, SNEES, and computation time for the twenty-orbits case 75

5.1 Initial conditions 103
5.2 RMSE for a single simulation 112
5.3 RMSE for 100 Monte Carlo simulations 115

6.1 Analysis of the weighing exponent in the mKLFCM 135
6.2 Analysis of the penalty parameter in the RKLFCM 135
6.3 Monte Carlo averaged JSD for Exemple 1 (100 simulations) 139
6.4 Monte Carlo averaged RMSE, NCI, and computation time for Example 2 (200 simulations) 143

List of Figures

3.1 Time averaged RMSE for 200 random realizations 35
3.2 Time history of the absolute NCI value for 200 random realizations 37
3.3 Time averaged effective sample size for 200 random realizations . . 37
3.4 Monte Carlo averaged RMSE for 100 random realizations 39
3.5 Time history of the absolute NCI value for 100 random realizations 40
3.6 Time averaged effective sample size for 100 random realizations . . 40
3.7 Monte Carlo averaged RMSE for 100 random realizations 44
3.8 Time history of the absolute NCI value for 100 random realizations 44

4.1 The flow chart of the modified EnGMF for orbit determination . . . 64
4.2 The average RMSE for 100 Monte Carlo simulations, 1 pass per orbit 68
4.3 The SNEES value and the average computation time per filtering run for 100 Monte Carlo simulations 69
4.4 Monte Carlo averaged RMSE and SNEES value for all the six cases 71
4.5 Monte Carlo averaged computation time per filtering run 72
4.6 The average RMSE for 100 Monte Carlo simulations, the ten-orbits case 72
4.7 The SNEES value and the average computation time per filtering run for 100 Monte Carlo simulations, the ten-orbits case 73
4.8 The average RMSE for 100 Monte Carlo simulations, the twenty-orbits case 74
4.9 The SNEES value and the average computation time per filtering run for 100 Monte Carlo Simulations, the twenty-orbits case 75

5.1 Geometry of reflection 84
5.2 Comparison of the sigma points and weights in UKF and TIUKF . . 97
5.3 Position, attitude, and surface parameter errors with the PFGMM without the constraint information 105
5.4 Position, attitude, and surface parameter errors with the PFGMM with the constraint information 107

5.5 Position, attitude, and surface parameter errors with the TIUKF . . 109
5.6 Position, attitude, and surface parameter errors with the TEKF with underweighting 110
5.7 Position, attitude, and surface parameter errors with the TEKF with considering surface parameters 111
5.8 Monte Carlo averaged RMSE of position and velocity for 100 random realizations 113
5.9 Monte Carlo averaged RMSE of attitude and angular velocity for 100 random realizations 114
5.10 Monte Carlo averaged RMSE of surface parameters for 100 random realizations 114
5.11 Time history of the absolute NCI value for 100 random realization . 116

6.1 Geometric interpretation of the new regularizer 131
6.2 Analysis of the weighing exponent in the mKLFCM 134
6.3 Analysis of the penalty parameter in the RKLFCM 136
6.4 Prior and measurement distribution for example 1 137
6.5 True posterior distribution for example 1 137
6.6 The PFGMM solutions with the four different clustering algorithms for a single simulation are presented with (a) the JSD value (the K-means) = 5.6130, (b) the JSD value (the EM algorithm for GMMs) = 5.4898, (c) the JSD value (the mKLFCM) = 4.4212, and (d) the JSD value (the RKLFCM) = 4.2193. 140
6.7 The average RMSE and the absolute NCI value for 200 Monte Carlo simulations 142

Chapter 1

Introduction

State estimation methods are widely used across a wide variety of applications in science and engineering fields including orbit determination [66, 98], robotics [11, 34], meteorology [4, 72], electric power systems [40, 69], and fluid dynamic systems [37, 42]. In particular, in orbit determination for space situational awareness (SSA), the importance of state estimation methods has been emphasized because of severe nonlinearities in orbit determination problems. The presence of these nonlinearities in orbital dynamics and measurements certainly makes the state estimation problem more challenging. Moreover, to date, there are more than 34,000 space objects (SOs) greater than 10 cm, such as used satellites and fragments of rocket, in Earth orbit [1], whereas only a limited number of sensors are available and used to estimate the states of SOs. As space debris becomes more congested due to the launching of new objects, it poses a serious threat to newly launched satellites. To solve this problem, extensive research has recently been done on improving the accuracy, consistency, and efficiency of state estimation [20, 21, 24, 50, 61, 65, 100].

In estimation theory, a state refers to information to explain the characteristics of a dynamic system, such as position, velocity, attitude, and angular velocity.

The dynamic system expresses the relation of time-varying states as a mathematical model with differential or difference equations. Unfortunately, it is nearly impossible to represent an exact dynamic system as a deterministic mathematical model because of modeling errors and disturbances. These uncertainties are usually regarded as noise and are expressed in a probability distribution. Because of the existence of these uncertainties, we use a measurement model and sensor data along with a dynamic model. In practice, however, not all states are directly measured due to monetary problems and/or physical constraints; therefore, state estimation methods are used to estimate and predict states with appropriate dynamic and measurement models based on noisy sensor data.

The objective of state estimation is to minimize the error between the true and estimated state of a system. State estimation is generally divided into a non-Bayesian (or frequentist) approach and a Bayesian approach [10]. Non-Bayesian estimation assumes that the true state is unknown but deterministic. The most popular examples of non-Bayesian estimators are least square (LS) and maximum likelihood (ML). On the other hand, Bayesian estimation is a stochastic state estimation approach where the posterior expectation is included in the cost function of an optimization problem; the goal of Bayesian estimation is to find the optimal solution of an unknown random variable. The most well-known Bayesian estimators are maximum a posteriori (MAP) and minimum means square error (MMSE).

In this dissertation, we focus on MMSE estimators whose optimal solution is the conditional mean [10]; MMSE estimators calculate the conditional probability density function (PDF) in a recursive way. For example, the process of stochastic

state estimation consists of two steps, a prediction step and an update step. The prediction step propagates the posterior distribution from the previous time step to the next time step. The propagated distribution is called the prior distribution. In the update step, the prior distribution is updated by measurement information through Bayes' rule to obtain the posterior distribution at the current time. In the following paragraph, a brief review of the types of MMSE estimators is introduced and, in subsequent chapters, these types are elaborated.

For linear systems with additive Gaussian noise, the Kalman filter is the optimal MMSE estimator [49]. The Kalman filter is first established for linear systems and can be extended to deal with nonlinear systems. Linear estimators for nonlinear systems are based on the linear MMSE (LMMSE) framework [56] and classified generally into two types according to the approximation methods used to estimate the first two moments mean and covariance matrix required for an LMMSE estimate. As extensions of the Kalman filter for nonlinear systems, the extended Kalman filter (EKF) [29], the Gaussian second-order filter (GSOF) [67], the iterated EKF [29], and the recursive update filter (RUF) [114] are defined with a local linear approximation. Another class of linear estimators for nonlinear systems rely on a set of deterministic regression points for statistical linearization of the nonlinear functions to obtain the first two moments [54]. These estimators include the quadrature Kalman filter (QKF) [6], the unscented Kalman filter (UKF) [46], and and the cubature Kalman filter (CKF) [5]. These linear estimators for nonlinear systems are also called linear Gaussian filters since they are derived under the assumption that all distributions are Gaussian, which is not guaranteed in practice.

Nonlinear filters such as the Gaussian sum filter (GSF) and particle filter (PF) are therefore used to deal with very nonlinear/non-Gaussian problems. The GSF deals with non-Gaussian distributions by obtaining the approximation of the PDF as a Gaussian mixture model (GMM) [2, 97]. Recently, there have been considerable efforts to improve the original GSF to better account for nonlinearities [19, 25, 26, 33, 85, 103, 106]. Other nonlinear estimators, such as PFs or sequential Monte Carlo methods, include the algorithms that approximate a PDF using a set of finite number of samples [7], such as the bootstrap particle filter (BPF) [30], the auxiliary particle filter (APF) [79], and the regularized particle filter (RPF) [92].

The objective of this dissertation is to develop new Bayesian approaches for highly nonlinear systems with a theoretical study on combining the benefits of the GSF and PF. This is achieved mainly in two ways: (1) every particle of PF is regarded as a Gaussian component to establish a GMM (Chapters 3 and 4) and (2) all particles are classified into some Gaussian mixture components using a clustering algorithm (Chapters 5 and 6).

Chapter 3 proposes a novel sequential Monte Carlo algorithm that samples from a GMM approximation of the posterior distribution. The conditional PDF is approximated by a weighted sum of Gaussian distributions in the GSF and by a weighted sum of Dirac delta functions in the PF, respectively. Moreover, the Dirac delta function can be mathematically defined as a Gaussian component with a collapsed zero covariance matrix. Motivated by this definition, each particle of the pre-prior distribution is treated as a Gaussian component with a zero covariance matrix and the GSF algorithm is then used to compute the posterior distribution.

Two alternative algorithms are also proposed to further improve the accuracy of the baseline algorithm. The first algorithm is designed to draw samples from a GMM as an importance distribution instead of as the true posterior. The second alternative algorithm is aimed to obtain a better GMM approximation for the pre-prior distribution by choosing nonzero covariances whose optimal value can be calculated to remove the bias in the sample covariance. Accuracy and consistency performance are evaluated by a Monte Carlo analysis and are also compared to the BPF and RPF in the numerical examples considered.

Chapter 4 suggests a modified kernel-based ensemble Gaussian mixture filtering (EnGMF) to produce fast and consistent orbit determination capabilities in a sparse measurement environment. In the EnGMF [3], kernel density estimation (KDE), which is a non-parametric technique to estimate the PDF of a random variable [93], is used to integrate a GSF with a PF. This algorithm computes the covariance matrix of each Gaussian component using the bandwidth parameter of a kernel function. Although the bandwidth parameter can be obtained using a data-driven method, Silverman's rule of thumb is proposed here to reduce the computational burden of KDE. By using equinoctial orbital elements instead of Cartesian coordinates, Silverman's rule of thumb can provide a near-optimal bandwidth parameter for orbit determination with sparse observation data, thus improving the filter's performance. Numerical simulations are performed to test and analyze the proposed algorithm compared to state-of-the-art approaches in terms of accuracy, consistency, and efficiency.

Chapter 5 proposes a new approach to nonlinear estimation combining a

GSF and PF using an expectation-maximization (EM) clustering algorithm in the context of space object (SO) tracking. In this method, the EM clustering algorithm groups several particles into each Gaussian mixture component at the prior time and then estimates the posterior distribution using the GSF algorithm. In addition to the nonlinear estimator, two modifications of the EKF and UKF are introduced to use the additional constraint information of, and deal with high nonlinearities of, a system of orbit determination in geostationary Earth orbits (GEO). The performance of the three algorithms are analyzed through numerical simulation with a challenging estimation problem of SO tracking in GEO.

Chapter 6 develops two novel clustering methods for PF with GMMs (PFGMM) proposed in Chapter 5. When the GSF and PF are combined, different clustering algorithms can be applied to the PFGMM. It is proved that, under the assumption of a perfect clustering scheme, the PFGMM's density converges in probability to the true filter density, implying that the filter performance depends mainly on the performance of a clustering method [83]. It is desirable for the GSF to have a small enough covariance matrix such that nonlinear measurement functions can be accurately approximated by linearization in the support of each component, which can be implemented by the K-means clustering algorithm in the PFGMM. Moreover, the better approximation prior distribution has, the better performance the GSF provides, which can be assured by the EM clustering algorithm for the PFGMM. Based on modifications of the fuzzy c-means (FCM) clustering algorithm, two different clustering algorithms are proposed. Both proposed clustering algorithms are designed to merge the advantages of the K-means and EM clustering

algorithm for better performance with the PFGMM; they simultaneously minimize the covariance of each of the GMM components and maximize the likelihood function. The performance comparison of the RPF and the PFGMM with the K-means, the EM, and the proposed clustering algorithms are done through a Monte Carlo analysis.

1.1 Contributions of the Dissertation

The following are novel contributions of this work:

- Chapter 3
  - A new sequential Monte Carlo algorithm is proposed that samples from a GMM approximation of the posterior distribution.
  - Two small modifications of the baseline algorithm are proposed to further improve its accuracy.
- Chapter 4
  - A modified EnGMF is introduced to produce fast and consistent orbit determination capabilities in a sparse measurement environment.
- Chapter 5
  - A new nonlinear estimation algorithm combining the PF and GSF using the EM clustering method is proposed to deal with the high nonlinearity and weak observability of a system.

  - Two alternative estimation techniques based on the UKF and EKF are presented.

- Chapter 6

  - Two novel clustering algorithms for the PFGMM are proposed to simultaneously minimize the covariance for each of the components of a GMM and maximize the likelihood function based on the FCM clustering algorithm.

Chapter 2

Background Material

Throughout this dissertation, we consider general discrete-time nonlinear dynamics and measurements. The dynamics is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{\nu}_k) \tag{2.1}$$

where k is the time step, $\boldsymbol{x}_k$ is an $n_{\boldsymbol{x}} \times 1$ vector, $\boldsymbol{f}_k$ is some nonlinear function and the process noise $\boldsymbol{\nu}_k$ is a zero mean, white sequence, independent from the initial distribution of $\boldsymbol{x}_0$ and possessing covariance matrix Q_k. The measurement is

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k \tag{2.2}$$

the measurement noise $\boldsymbol{\eta}_k$ is a zero mean, white sequence with covariance matrix R_k, independent from all other random quantities.

2.1 Linear Filters for Nonlinear Systems

This section serves to introduce the concepts and algorithms of the EKF and UKF, which are based on the LMMSE estimator's framework for nonlinear systems.

2.1.1 The Extended Kalman Filter

The EKF is a nonlinear approximation of the Kalman filter that can be applied to nonlinear systems using the same Kalman filtering framework [29]. Given the system model, Eq. (2.1) and Eq. (2.2), the time update equations are described

as follows:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_{k|k}) \tag{2.3}$$

$$P^{xx}_{k+1|k} = F_k P^{xx}_{k|k} F_k^{\mathrm{T}} + G_k Q_k G_k \tag{2.4}$$

$$F_k = \left.\frac{\partial \boldsymbol{f}_k(\boldsymbol{x}, \boldsymbol{\nu})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k}} \tag{2.5}$$

$$G_k = \left.\frac{\partial \boldsymbol{f}_k(\boldsymbol{x}, \boldsymbol{\nu})}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_k} \tag{2.6}$$

where $\hat{\boldsymbol{x}}_{k+1|k}$ is the *a priori* state estimate, $P^{xx}_{k+1|k}$ is the *a priori* state estimation error covariance, F_k and G_k are the Jacobian of the dynamics evaluated at the posterior mean $\hat{\boldsymbol{x}}_{k|k}$ and at $\boldsymbol{\nu}_k = \mathbf{0}$, respectively, and $P^{xx}_{k|k}$ is the posterior state estimation error covariance at time instant k. The measurement update equations are:

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + K_{k+1}\left(\boldsymbol{y}_{k+1} - \boldsymbol{h}_{k+1}(\hat{\boldsymbol{x}}_{k+1|k})\right) \tag{2.7}$$

$$P^{xx}_{k+1|k+1} = P^{xx}_{k+1|k} - K_{k+1} W_{k+1} K_{k+1}^{\mathrm{T}} \tag{2.8}$$

$$K_{k+1} = P^{xx}_{k+1|k} H_{k+1}^{\mathrm{T}} W_{k+1}^{-1} \tag{2.9}$$

$$W_{k+1} = H_{k+1} P^{xx}_{k+1|k} H_{k+1}^{\mathrm{T}} + R_{k+1} \tag{2.10}$$

$$H_{k+1} = \left.\frac{\partial \boldsymbol{h}_{k+1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}} \tag{2.11}$$

where $P^{xx}_{k+1|k+1}$ is the *a posteriori* state estimation error covariance at time instant $k+1$, H_{k+1} is the Jacobian of the measurement evaluated at the prior mean $\hat{\boldsymbol{x}}_{k+1|k}$, K_{k+1} is the Kalman gain, and W_{k+1} is the measurement residual covariance.

2.1.2 The Unscented Kalman Filter

The UKF approximates nonlinear functions with statistical linearization using a set of sigma points [54]. The most common schemes to effectively calculate

sigma points are to assume that all distributions are Gaussian [47]. Given an $n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}$ error covariance matrix $P^{xx}_{k|k}$, we generate $2n_{\boldsymbol{x}} + 1$ sigma points as follows:

$$\mathcal{X}_{k|k} = \hat{\boldsymbol{x}}_{k|k} \mathbb{1}_{1\times(2n_{\boldsymbol{x}}+1)} + \left[\mathbf{0}_{n_{\boldsymbol{x}}\times 1},\ \sqrt{n_{\boldsymbol{x}} + \lambda} S_k\right] \tag{2.12}$$

where $\mathbb{1}_{1\times(2n_{\boldsymbol{x}}+1)}$ is an indicator function,

$$S_k = \left[\left(P^{xx}_{k|k}\right)^{1/2} - \left(P^{xx}_{k|k}\right)^{1/2}\right], \tag{2.13}$$

$\left(P^{xx}_{k|k}\right)^{1/2}$ is the square root of the $P^{xx}_{k|k}$ such that $P^{xx}_{k|k} = \left(P^{xx}_{k|k}\right)^{1/2}\left(P^{xx}_{k|k}\right)^{1/2(\mathrm{T})}$, and $\lambda = \alpha^2(n_{\boldsymbol{x}} + \kappa) - n_{\boldsymbol{x}}$ is a scaling parameter [107]. The parameter α tunes the spread of the sigma points around $\hat{\boldsymbol{x}}_{k|k}$ and it is usually set to a small positive number ($10^{-4} \leq \alpha \leq 1$). κ is a secondary scaling parameter which is usually set to $3 - n_{\boldsymbol{x}}$. Based on the above sigma points, the corresponding weights are calculated as follows:

$$W^m_0 = \frac{\kappa}{n_{\boldsymbol{x}} + \kappa}, \quad W^c_0 = \frac{\kappa}{n_{\boldsymbol{x}} + \kappa} + (1 - \alpha^2 + \beta) \tag{2.14}$$

$$W^m_j = W^c_j = \frac{0.5}{n_{\boldsymbol{x}} + \kappa}, \quad \text{for } j = 1, \cdots, 2n_{\boldsymbol{x}} \tag{2.15}$$

where the parameter β is used to include prior knowledge of the distribution of $\boldsymbol{x}$. With the above sigma points and weights, the time update equations are expressed as follows:

$$\mathcal{X}_{j,k+1|k} = \boldsymbol{f}_k(\mathcal{X}_{j,k|k}), \quad j = 0, \cdots, 2n_{\boldsymbol{x}} \tag{2.16}$$

$$\hat{\boldsymbol{x}}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W^m_j \mathcal{X}_{j,k+1|k} \tag{2.17}$$

$$P^{xx}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W^c_j \left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right]\left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right]^{\mathrm{T}} + Q_k \tag{2.18}$$

Using the propagated estimates $\hat{\boldsymbol{x}}_{k+1|k}$ and $P^{xx}_{k+1|k}$, a new set of sigma points $\mathcal{X}_{k+1|k}$ and the corresponding weights are recalculated. The measurement update equations are then expressed as follows:

$$\mathcal{Y}_{j,k+1|k} = \boldsymbol{h}_{k+1}(\mathcal{X}_{j,k+1|k}), \quad j = 0, \cdots, 2n_{\boldsymbol{x}} \tag{2.19}$$

$$\hat{\boldsymbol{y}}_{k+1} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^m \mathcal{Y}_{j,k+1|k} \tag{2.20}$$

$$P^{yy}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^c \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right] \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right]^{\mathrm{T}} + \boldsymbol{R}_{k+1} \tag{2.21}$$

$$P^{xy}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^c \left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right] \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right]^{\mathrm{T}} \tag{2.22}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + P^{xy}_{k+1|k} \left(P^{yy}_{k+1|k}\right)^{-1} \left(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}\right) \tag{2.23}$$

$$P^{xx}_{k+1|k+1} = P^{xx}_{k+1|k} - P^{xy}_{k+1|k} \left(P^{yy}_{k+1|k}\right)^{-1} \left(P^{xy}_{k+1|k}\right)^{\mathrm{T}} \tag{2.24}$$

where $P^{yy}_{k+1|k}$ is the measurement residual covariance, $P^{xy}_{k+1|k}$ is the cross covariance.

The linear filters for nonlinear systems are also called linear Gaussian filters since they are derived under the assumption that all distributions are Gaussian. However, in practice, nonlinear systems does not guarantee Gaussian distributions. Therefore, nonlinear filters such as the GSF and PF are used to deal with nonlinear/non-Gaussian problems.

2.2 Nonlinear Filters for Nonlinear Systems

The GSF and PF are two common solutions to the nonlinear Bayesian estimation problem, and they are briefly reviewed in this section.

2.2.1 The Gaussian Sum Filter

The GSF approximates the conditional PDF by combining several Gaussian components having different means and covariance matrices, and this approximation of the probability distribution is called a GMM [2, 97]. The conditional PDF of $\boldsymbol{x}_k|\boldsymbol{y}_1...\boldsymbol{y}_k$ is expressed as follows:

$$p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) = \sum_{i=1}^{N} \omega_{k|k}^{(i)}\, n(\boldsymbol{x}_k;\ \boldsymbol{\mu}_{k|k}^{(i)}, \boldsymbol{P}_{k|k}^{(i)}) \tag{2.25}$$

where $n(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{P})$ represents the Gaussian pdf with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{P}$, and $\omega_{k|k}^{(i)}$, $\boldsymbol{\mu}_{k|k}^{(i)}$ and $\boldsymbol{P}_{k|k}^{(i)}$ are the weights, means, and covariance matrices of the i-th Gaussian component. The PDF's normalization and positivity properties lead to the following constraints on the weights

$$\omega_{k|k}^{(i)} \geq 0,\ \forall i \qquad \sum_{i=1}^{N} \omega_{k|k}^{(i)} = 1 \tag{2.26}$$

(It is actually possible to define some of the weights negative, but that type of GMM approximation is not considered here.) Assuming the covariance matrices are "small" enough (such that linearization of the dynamics and measurements holds in the domain of likely realization of each of the components), then each of the components remains approximately Gaussian at all times and it is propagated and updated using the conventional EKF (or UKF) equations. The time update equations are described as:

$$\boldsymbol{\mu}_{k+1|k}^{(i)} = \boldsymbol{f}_k\big(\boldsymbol{\mu}_{k|k}^{(i)}\big) \tag{2.27}$$

$$\boldsymbol{P}_{k+1|k}^{(i)} = \boldsymbol{F}_k^{(i)} \boldsymbol{P}_{k|k} \boldsymbol{F}_k^{(i)\mathrm{T}} + \boldsymbol{G}_k^{(i)} \boldsymbol{Q}_k \boldsymbol{G}_k^{(i)\mathrm{T}} \tag{2.28}$$

$$\omega_{k+1|k}^{(i)} = \omega_{k|k}^{(i)} \tag{2.29}$$

where $\boldsymbol{F}_k^{(i)}$ and $\boldsymbol{G}_k^{(i)}$ are the Jacobian of the dynamics evaluated at the component's mean $\boldsymbol{x}_x = \boldsymbol{\mu}_{k|k}^{(i)}$ and at $\boldsymbol{\nu}_k = \boldsymbol{0}$, respectively.

The measurement update follows from Bayes's rule and is given by

$$\boldsymbol{\mu}_{k|k}^{(i)} = \boldsymbol{\mu}_{k|k-1}^{(i)} + \boldsymbol{K}_k^{(i)} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k \big(\boldsymbol{\mu}_{k|k-1}^{(i)} \big) \Big) \tag{2.30}$$

$$\boldsymbol{P}_{k|k}^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)} - \boldsymbol{K}_k^{(i)} \left(\boldsymbol{H}_k^{(i)} \boldsymbol{P}_{k|k-1}^{(i)} \boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \right) \boldsymbol{K}_k^{(i)\mathrm{T}} \tag{2.31}$$

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)} \, \boldsymbol{H}_k^{(i)\mathrm{T}} \, \left(\boldsymbol{H}_k^{(i)} \boldsymbol{P}_{k|k-1}^{(i)} \boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \right)^{-1} \tag{2.32}$$

$$\omega_{k|k}^{(i)} = \frac{\omega_{k|k-1}^{(i)} \, \beta_k^i}{\sum_{i=1}^{N} \omega_{k|k-1}^{(i)} \, \beta_k^i} \tag{2.33}$$

where

$$\beta_k^i = n\Big(\boldsymbol{y}_k;\, \boldsymbol{h}_k \big(\boldsymbol{x}_{k|k-1}^{(i)} \big),\, \boldsymbol{H}_k^{(i)} \boldsymbol{P}_{k|k-1}^{(i)} \boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \Big) \tag{2.34}$$

where $\boldsymbol{H}_k^{(i)}$ is the Jacobian of the measurement evaluated at the prior mean $\boldsymbol{\mu}_{k|k-1}^{(i)}$. The weights are scaled so that they add to one.

Lastly, the total mean $\boldsymbol{\mu}_{k|k}$ and covariance matrix $\boldsymbol{P}_{k|k}$ of the posterior GMM are given by

$$\boldsymbol{\mu}_{k|k} = \sum_{i=1}^{N} \omega_{k|k}^{(i)} \, \boldsymbol{\mu}_{k|k}^{(i)} \tag{2.35}$$

$$\boldsymbol{P}_{k|k} = \sum_{i=1}^{N} \omega_{k|k}^{(i)} \left(\boldsymbol{P}_{k|k}^{(i)} + \boldsymbol{\mu}_{k|k}^{(i)} \boldsymbol{\mu}_{k|k}^{(i)\mathrm{T}} - \boldsymbol{\mu}_{k|k} \boldsymbol{\mu}_{k|k}^{\mathrm{T}} \right) \tag{2.36}$$

It is noted that the GMM approximation of the conditional PDF approaches to the true PDF under the assumption that there are a sufficient number of Gaussian components and that each of them has covariance matrix small enough such that the

linearization of each component around its mean is representative of the nonlinear dynamics and measurements.

2.2.2 The Particle Filter

Particle filters are a subset of sequential Monte Carlo methods that use sequential importance sampling with resempling (SISR). The PF approximates the continuous PDF as a discrete probability mass function (PMF), therefore the PDF is composed by a weighted sum of Dirac delta functions [7].

$$p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) \approx \sum_{i=1}^{N} \omega_{k|k}^{(i)}\, \delta(\boldsymbol{x}_k - \boldsymbol{x}_{k|k}^{(i)}) \tag{2.37}$$

If it were feasible to compute the actual posterior distribution at the next time step $p_{\boldsymbol{x}_{k+1}}(\boldsymbol{x}_{k+1})$ starting from $p_{\boldsymbol{x}_k}(\boldsymbol{x}_k)$ and to sample from it; then we would use standard Monte Carlo techniques. However, since it is usually unfeasible to sample from the actual posterior distribution, an importance distribution is often used instead. The bootstrap particle filter (BPF) uses the transition distribution as the importance distribution. With capital letters we indicate the collection of all random vectors identified by the corresponding lower case letter, up to and including the current time.

$$\boldsymbol{X}_k = \boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_k, \quad \boldsymbol{Y}_k = \boldsymbol{y}_1, \cdots, \boldsymbol{y}_k \tag{2.38}$$

Then, the BPF importance distribution is given by

$$\pi(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k, \boldsymbol{Y}_{k+1}) = p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \tag{2.39}$$

the sample $\boldsymbol{x}_{k+1}^{(i)}$ is obtained by first sampling $\boldsymbol{\nu}_k^{(i)}$ from the process noise. In this work the samples $\boldsymbol{\nu}_k^{(i)}$ are drawn from a Gaussian distribution and $\boldsymbol{x}_{k+1}^{(i)}$ are obtained

as follows:

$$\boldsymbol{x}_{k+1}^{(i)} = \boldsymbol{f}_k(\boldsymbol{x}_k^{(i)}, \boldsymbol{\nu}_k^{(i)}) \tag{2.40}$$

The importance weights are then calculated as

$$\omega_{k+1}^{(i)} \propto \omega_k^{(i)}\ p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}^{(i)}) = \omega_k^{(i)}\ n\Big(\boldsymbol{y}_{k+1};\ \boldsymbol{h}_k\big(\boldsymbol{x}_{k+1}^{(i)}\big),\ \boldsymbol{R}_{k+1}\Big) \tag{2.41}$$

Sample impoverishment is common in the BPF, and the weights update step is usually followed by a resampling step.

Chapter 3

Sequential Monte Carlo Filtering with Gaussian Mixture Sampling

3.1 Comparison with State of the Art

Bayesian stochastic estimation of nonlinear and non-Gaussian dynamical systems using sequential Monte Carlo methods continues to receive considerable attention in the literature [8, 23, 90]. The Kalman filter provides an exact solution to the minimum mean-square error estimation problem for linear systems corrupted by additive Gaussian noise [49]. However, in practice, the conditions for optimality of the Kalman filter are easily and often violated. The EKF is a non-optimal approximation of the optimal Kalman filter that can be applied to nonlinear systems using the same Kalman filtering framework [29]. The possible divergence of the EKF estimates due to severe nonlinearities is a drawback of this procedure. Other linear estimators of nonlinear systems include algorithms that rely on a set of deterministic regression points [54], such as the QKF [6], the UKF [46], and the CKF [5]. These algorithms employ the Gaussian approximation and statistical linearization of the nonlinear functions through a set of regression points. However, these methodologies are not always feasible for very high nonlinearities when the state's PDF is multimodal or very non-Gaussian.

The GSF is a nonlinear estimator for nonlinear systems [2, 97]. It is able to account for large deviations from Gaussianity and accommodate multimodal distri-

This chapter is based on: Sehyun Yun and Renato Zanetti, "Sequential Monte Carlo Filtering with Gaussian Mixture Sampling," *Journal of Guidance, Control, and Dynamics*, 42(9), pp.2069-2077, 2019. The primary content in this work was contributed by the first author.

butions by approximating the non-Gaussian PDF as a GMM. The GSF includes one linear estimator, such as EKF or UKF, for each of the GMM components. The GSF works best when enough components are taken, each of which with a small enough covariance matrix such that the nonlinear functions can accurately be linearized in the support of each of the components. Much recent research exists in improving the original GSF algorithm to better adapt to, and account for, nonlinearities [19, 25, 26, 33, 85, 103, 106].

While GSFs approximate the PDFs as a sum of Gaussians, sequential Monte Carlo methods approximate them by discretization using a finite number of random samples. Monte Carlo methods need to draw from the actual distributions, which are often arduous to obtain; sequential importance sampling (SIS) algorithms, on the other hand, sample from an importance sampling distribution and adjust the weights of each sample accordingly. Particle filters are a family of SIS algorithms that include a resampling step to mitigate particle (i.e., sample) degeneracy [7]. One of the most popular algorithms chooses the importance distribution as the transition distribution, the BPF [30]. One possible drawback of the BPF is that it does not directly account for the value of the measurement in the sampling distribution. The APF mitigates this issue by using an auxiliary variable to account for the value of the measurement in the importance distribution [79]. The resampling step is often critical for practical uses of the PF and is usually done sampling from a discrete distribution. The RPF draws from a continuous distribution approximation of the PDF [23] by perturbing the particles after resampling to add diversity to the state space. The approach presented in this chapter contains a new methodology that

includes both these improvements: it samples from a continuous distribution that incorporates the contributions of the current measurement.

The PF approximates distributions as discrete, i.e., as weighted sum of Dirac deltas. Other approaches to discretize the PDF include the point mass filter (PMF) [15], deterministic Dirac mixtures with equal weights [92] and to combine particle filters with GMMs. In Refs. [82] and [81], the authors start from a GMM and at each cycle they resample in a manner similar to a particle filter. Their resampling step is subject to a matrix inequality constraint that ensures the covariance of each of the resampled Gaussian components stays below a desired tunable value.

Ref. [52] starts from the Gaussian particle filter derived in [53] to build the Gaussian sum particle filter (GSPF). The GSPF is basically a bank of Gaussian particle filters approximating the conditional distributions by weighted Gaussian mixtures. Ref. [83] introduces the particle Gaussian mixture filter (PGMF) and employs an ensemble of randomly sampled states for the propagation of the conditional state probability density. The propagated ensemble is clustered to recover a GMM representation of the propagated PDF. Finally, the posterior PDF can be obtained through a GSF update. This approach is somewhat reminiscent of the RPF, which uses kernel density estimation as the clustering algorithm.

In this chapter, we propose to always sample from the posterior distribution, never to combine the distribution at the prior time with an importance distribution, as done in SIS. Therefore, the methodology and the algorithms derived in this work are conceptually and practically very different from the GSPF and PGMF. The proposed methodology is also different from the RPF, since the RPF employs kernel

density estimation on the particles in order to resample from a continuous distribution. In this work we calculate the posterior distribution directly, we do not approximate it via clustering or kernel density estimator starting from samples. We do not utilize sequential importance sampling like in the GSPF. Moreover, no particles propagation and clustering occurs like in the PGMF, rather an initial GMM is generated at each cycle and from it a posterior PDF is obtained. In addition to the main result, two modifications of the baseline algorithm are proposed to further improve its accuracy. First, an importance sampling version of the algorithm is developed. Then, in the second modification, the initial covariance of the GMM components is not set to zero, but to a small value that removes the bias in the sample covariance.

The remainder of this chapter is organized as follows: Section 3.2 introduces the new algorithms. Then, in Section 3.3, simulation results using the proposed algorithm are presented followed by some concluding remarks on the methodology and results.

3.2 Algorithms Development

At each time step, sequential Monte Carlo methods in general, and particle filters in particular, necessitate to start from a good set of samples that accurately and sufficiently represent the true distribution. Assuming such an initial set of samples exists, our goal is to approximate the distribution of $\boldsymbol{x}_k|\boldsymbol{Y}_k$ using sequential Monte Carlo methods. We will present one main algorithm and then show two small modifications to it. In a Monte Carlo method, ideally we would want to sample from $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$, and the first algorithm we propose does exactly that.

We assume that we have a good representation of the distribution at the prior time, that is to say, we have a set of of N independent and identically distributed (i.i.d.) samples $\boldsymbol{x}_{k-1}^{(i)}$ such that

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{3.1}$$

the basic idea behind the proposed algorithm is that the Dirac delta function $\delta(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)$ is the limit as the covariance matrix goes to zero of a Gaussian distribution with mean $\bar{\boldsymbol{x}}_k^{(i)}$

$$\delta(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) = n(\boldsymbol{x}_k;\ \bar{\boldsymbol{x}}_k, \boldsymbol{O}) \tag{3.2}$$

3.2.1 Algorithm I - Sampling from a GMM Posterior

As mentioned above we start from N i.i.d. samples of $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1})$ and we interpret the discretized distribution as a GMM

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) = \sum_{i=1}^{N} \frac{1}{N} n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{O}) \tag{3.3}$$

We can therefore propagate this distribution forward in time with the GSF equations to obtain

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N}\ n(\boldsymbol{x}_k; \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}), \boldsymbol{G}_{k-1}^{(i)} \boldsymbol{Q}_{k-1} \boldsymbol{G}_{k-1}^{(i)\mathrm{T}}) \tag{3.4}$$

We can then process the measurement to obtain

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N} \omega_k^{(i)}\, n(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}) \tag{3.5}$$

$$\boldsymbol{\mu}_k^{(i)} = \boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big) + \boldsymbol{K}_k\Big(\boldsymbol{y}_k - \boldsymbol{h}_k\big(\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big)\big)\Big) \tag{3.6}$$

$$\boldsymbol{P}_k^{(i)} = \boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}} - \boldsymbol{K}_k^{(i)}\boldsymbol{W}_k^{(i)}\boldsymbol{K}_k^{(i)\mathrm{T}} \tag{3.7}$$

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}\, \boldsymbol{H}_k^{(i)\mathrm{T}}\, (\boldsymbol{W}_k^{(i)})^{-1} \tag{3.8}$$

$$\boldsymbol{W}_k^{(i)} = \boldsymbol{H}_k^{(i)}\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}\boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \tag{3.9}$$

$$\boldsymbol{G}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}\big(\boldsymbol{x}, \boldsymbol{\nu}\big)}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_{k-1}^{(i)}} \tag{3.10}$$

$$\boldsymbol{H}_k^{(i)} = \left.\frac{\partial \boldsymbol{h}_k\big(\boldsymbol{x}\big)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{f}_{k-1}\left(\boldsymbol{x}_{k-1}^{(i)}\right)} \tag{3.11}$$

$$\omega_k^{(i)} \propto n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k\big(\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big)\big),\ \boldsymbol{W}_k^{(i)}\Big) \tag{3.12}$$

where the weights in Eq. (3.12) are normalized.

We can now sample from the GMM distribution in Eq. (3.5) to obtain N i.i.d. samples of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$; from these samples we can construct a Bayesian estimate and we can use them as a starting point for the next iteration.

To draw from a GMM we follow these steps:

1. Draw N samples $u^{(i)}$ from a uniform distribution between 0 and 1

2. For each i, find the index ℓ_i (where the subscipt i is to reinforce the fact there is one index for each value of $i = 1...N$) such that $\sum_{j=1}^{\ell_i - 1} \omega^{(j)} < u^{(i)} \leq \sum_{j=1}^{\ell_i} \omega^{(j)}$, where we define $\sum_{j=1}^{0} \omega^{(i)} = 0$

3. Draw $\boldsymbol{x}_k^{(i)}$ from the Gaussian distribution $n(\boldsymbol{x}_k;\ \boldsymbol{m}_{k|k-1}^{(\ell_i)}, \boldsymbol{P}_{k|k-1}^{(\ell_i)})$

Our approach of sampling from the GMM has two benefits. First, components with small weight are unlikely to produce a sample, therefore the resampling step is effectively already included in the sampling step. In the BPF, the process noise provides sample diversity after resampling. In a GMM the sample diversity is obtained directly since the Gaussian components are continuous distributions that already contain the contribution of the process noise. Other algorithms such as the RPF need to perform additional steps starting from the discrete distribution to obtain a continuous distribution to resample from. The second benefit of this algorithm is that, unlike the BPF, the GMM distribution accounts for the value of the measurement $\boldsymbol{y}_k$. This approach is reminiscent of the APF, except that the full Bayes update is performed which allows us to directly sample rather than doing importance sampling.

Our proposed approach provides very good performance if:

1. The process noise covariance is not large enough such that the linearization of the measurement function $\boldsymbol{h}(\boldsymbol{x})$ is invalid in a region around $\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{0}\big)$ whose spread is consistent with $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$

2. The number of samples we start from is sufficient to accurately approximate the distribution at the prior time: $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{O})$

This second assumption is common to all particle filters. If one of these two assumptions fail, the same algorithm proposed here can be used with the following

mitigation strategies: 1) expressing the process noise itself as a GMM such that each component has a small enough covariance and 2) drawing more points from $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ as a starting point of the algorithm. However, at the cost of more computations, it is possible to mitigate these two issues in alternative ways as presented in the following two subsections.

3.2.2 Algorithm II - Importance Sampling from a GMM Posterior

If the distribution at the prior time is accurate, but we have reason to believe our GMM approximation of the distribution at the current time in Eq. (3.5) is not as accurate, it is possible to draw from the GMM in Eq. (3.5) as an importance distribution rather than as the true posterior. When drawing samples $\boldsymbol{x}_k^{(i)}$ from an importance distribution $\pi(\boldsymbol{x}_k|\boldsymbol{Y}_k)$, it is necessary to compute the true probability density $p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)$ in order to compute the importance weights. Therefore, we still need good knowledge of $p(\boldsymbol{x}_{k-1}^{(i)}|\boldsymbol{Y}_{k-1})$. Algorithm II proposed in this subsection provides a good methodology when we have enough samples $\boldsymbol{x}_{k-1}^{(i)}$ to accurately represent $p(\boldsymbol{x}_{k-1}^{(i)}|\boldsymbol{Y}_{k-1})$, but the GMM approximation of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ is not sufficiently accurate. This situation can occur when the linearization assumption taken by each of the components of the GMM is not accurate, such as when the nonlinearities of the measurement function $\boldsymbol{h}_k(\boldsymbol{x}_k)$ are significant in a region around $\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{0})$ spanned by the likely realizations of the component.

Assume the following approximation of the true posterior is more accurate

than Eq. (3.5):

$$p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) \int p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})\, d\boldsymbol{x}_{k-1} \tag{3.13}$$

$$\approx p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) \sum_{j=1}^{N} p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(j)})\, p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1}) \tag{3.14}$$

$$= n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k\big(\boldsymbol{x}_k^{(i)}\big), \boldsymbol{R}_k\Big) \sum_{j=1}^{N} \zeta_{k-1}^{(j)}\, n\Big(\boldsymbol{x}_k^{(i)};\ \boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(j)}\big), \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}}\Big) \tag{3.15}$$

where the prior distribution $p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1})$ no longer has all weights equal to $1/N$:

$$p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1}) = \sum_{j=1}^{N} \zeta_{k-1}^{(j)}\, n\Big(\boldsymbol{x}_k^{(i)};\ \boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{O}\Big) \tag{3.16}$$

then we can use Eq. (3.5) as the importance density

$$\pi(\boldsymbol{x}_k|\boldsymbol{Y}_k) = \sum_{i=1}^{N} \omega_k^{(i)}\, n\big(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}\big) \tag{3.17}$$

where $\omega_k^{(i)}$ are the weights of the i-th Gaussian component as defined in Eq. (3.12). The importance weights are calculated as

$$\xi_k^{(i)} = \frac{p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)} \tag{3.18}$$

$$\zeta_k^{(i)} = \frac{\xi_k^{(i)}}{\sum_{i=1}^{N} \xi_k^{(i)}} \tag{3.19}$$

The posterior density is therefore approximated as

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N} \zeta_k^{(i)}\, \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) = \sum_{i=1}^{N} \zeta_k^{(i)}\, n(\boldsymbol{x}_k; \boldsymbol{x}_k^{(i)}, \boldsymbol{O}) \tag{3.20}$$

notice that at the start of each iteration the initial weights $\zeta_k^{(i)}$ of the GMM are not $1/N$ as in Algorithm 1.

For large N, Algorithm II can be significantly more computationally expensive than Algorithm I because of the summation in Eq. (3.16) that is performed for each sample. In other words, Algorithm II has complexity of order N^2. Therefore, in situations where the process noise covariance is large, expressing the process noise itself as a GMM and using Algorithm I is possibly preferable from a computational standpoint.

The pre-update error covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ being too small or not full rank could lead to particle impoverishment issues (all particle filters suffer from this problem). To overcome this, the following algorithm which uses nonzero initial covariances is proposed.

3.2.3 Algorithm III - Estimation with Non-Zero Initial Covariance

A better GMM approximation of $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ than Eq. (3.3) can be obtained by choosing nonzero covariances $\boldsymbol{P}_{k-1}^{(i)}$ for each of the components

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N}\, n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{P}_{k-1}^{(i)}) \tag{3.21}$$

Calculation of optimal values of $\boldsymbol{P}_{k-1}^{(i)}$ (for example minimizing the L2 norm of the difference between PDFs) is often infeasible or computationally expensive; a very simple alternative approach is to remove the bias in the sample covariance as described in this section.

When all the weights are the same, the covariance matrix of the GMM dis-

tribution is given by

$$\boldsymbol{P}_{k-1}^{GMM} = \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{P}_{k-1}^{(i)} + \boldsymbol{x}_{k-1}^{(i)}\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}} \tag{3.22}$$

$$\boldsymbol{\mu}_{k-1} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{k-1}^{(i)} \tag{3.23}$$

When $\boldsymbol{P}_{k-1}^{(i)} = \boldsymbol{O}$, this reduces to

$$\begin{aligned}\boldsymbol{P}_{k-1}^{BIAS} &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{x}_{k-1}^{(i)}\,\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}}\end{aligned} \tag{3.24}$$

which is a biased estimator of the covariance matrix since it is, on average, too small. An unbiased estimator of the covariance matrix is

$$\boldsymbol{P}_{k-1}^{UNB} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{3.25}$$

A very simple method to choose a nonzero value for the covariance matrix of the components in Eq. (3.21) is to choose $\boldsymbol{P}_{k-1}^{(i)}$ such that the GMM covariance matrix is unbiased

$$\begin{aligned}&\frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{P}_{k-1}^{(i)} + \boldsymbol{x}_{k-1}^{(i)}\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}} \\ &\quad = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}}\end{aligned} \tag{3.26}$$

Assuming all samples have the same covariance matrix, the solution to this equation is given by

$$\begin{aligned}\boldsymbol{P}_{k-1}^{(i)} &= \frac{1}{N(N-1)}\sum_{j=1}^{N}\left(\boldsymbol{x}_{k-1}^{(j)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(j)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \frac{1}{N}\boldsymbol{P}_{k-1}^{UNB} \qquad \forall i\end{aligned} \tag{3.27}$$

For example, if we have a scalar state with unbiased sample variance of $P_{k-1}^{UNB} = 1$ and we choose to draw 100 samples, each of the 100 components of the GMM will have standard deviation $\sigma_{k-1}^{(i)} = \sqrt{P_{k-1}^{(i)}} = 0.1$.

The remainder of the algorithm is similar to Algorithm I

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N} \omega_k^{(i)}\ n(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}) \tag{3.28}$$

$$\boldsymbol{\mu}_k^{(i)} = \boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big) + \boldsymbol{K}_k\Big(\boldsymbol{y}_k - \boldsymbol{h}_k\big(\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big)\big)\Big) \tag{3.29}$$

$$\boldsymbol{P}_k^{(i)} = \boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}} - \boldsymbol{K}_k^{(i)}\boldsymbol{W}_k^{(i)}\boldsymbol{K}_k^{(i)\mathrm{T}} \tag{3.30}$$

$$\boldsymbol{K}_k^{(i)} = (\boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}})\ \boldsymbol{H}_k^{(i)\mathrm{T}}\ (\boldsymbol{W}_k^{(i)})^{-1} \tag{3.31}$$

$$\boldsymbol{W}_k^{(i)} = \boldsymbol{H}_k^{(i)}(\boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}})\boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \tag{3.32}$$

$$\boldsymbol{F}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}\big(\boldsymbol{x}, \boldsymbol{\nu}\big)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{k-1}^{(i)}} \tag{3.33}$$

$$\boldsymbol{G}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}\big(\boldsymbol{x}, \boldsymbol{\nu}\big)}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_{k-1}^{(i)}} \tag{3.34}$$

$$\boldsymbol{H}_k^{(i)} = \left.\frac{\partial \boldsymbol{h}_k\big(\boldsymbol{x}\big)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{f}_{k-1}\left(\boldsymbol{x}_{k-1}^{(i)}\right)} \tag{3.35}$$

$$\omega_k^{(i)} \propto n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k\big(\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}\big)\big),\ \boldsymbol{W}_k^{(i)}\Big) \tag{3.36}$$

The weights in Eq. (3.36) are normalized, and we can now sample from this GMM distribution to obtain N i.i.d. samples of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$.

In Algorithm III, we calculate the actual posterior distribution as a GMM and sample directly from it. Moreover, the covariance matrix of the components $\boldsymbol{P}_{k-1}^{(i)}$ is calculated, which makes Algorithm III practically and conceptually different from the RPF, in that the covariance is not merely used for particle resampling.

3.3 Numerical Results

In order to evaluate the algorithms proposed in this chapter, four different examples are considered: a simple motivating example, the univariate nonstationary growth model (used in [30, 51, 52]), a Lorenz96 system (used in [74, 83]), and the blind tricyclist problem (used in [80–82]).

3.3.1 Single Step Example

Consider the following simple motivating example. A bivariate normal random vector $\boldsymbol{x}_0$ is distributed as

$$\boldsymbol{x}_0 \sim n\left(\boldsymbol{x}_0;\ \boldsymbol{\mu}_0,\ \boldsymbol{P}_0\right) = n\left(\boldsymbol{x}_0;\ \begin{bmatrix}-3\\0\end{bmatrix},\ \begin{bmatrix}7.2 & 0\\0 & 21.6\end{bmatrix}\right) \tag{3.37}$$

and evolves as

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{\nu} \tag{3.38}$$

where

$$\boldsymbol{\nu} \sim n\left(\boldsymbol{\nu};\ \mathbf{0},\ \boldsymbol{Q}\right) = n\left(\boldsymbol{\nu};\ \begin{bmatrix}0\\0\end{bmatrix},\ \begin{bmatrix}0.2 & 0\\0 & 0.2\end{bmatrix}\right) \tag{3.39}$$

A measurement is available and given by

$$y = \|\boldsymbol{x}_1\| + \eta \tag{3.40}$$

where

$$\eta \sim n\left(\eta;\ 0,\ R\right) = n\left(\eta;\ 0,\ 0.01\right) \tag{3.41}$$

We start from $N = 300$ independently drawn samples of $\boldsymbol{x}_0$ and we apply the BPF, the APF, Algorithm I (A1) and Algorithm III (A3) from this chapter.

For the BPF we draw N independent samples from $\boldsymbol{\eta}$, and update the weights as

$$w_{BPF}^{(i)} \propto n(\boldsymbol{\nu}^{(i)};\ \mathbf{0}, \boldsymbol{Q})\ n(\|\boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}\|;\ 0, R) \tag{3.42}$$

The effective number of particles is calculated as

$$N_{eff}^{BPF} = \frac{1}{\sum_{i=1}^{N} (w_{BPF}^{(i)})^2} \tag{3.43}$$

and $\boldsymbol{x}_1^{(i),BPF} = \boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}$ with associated weight $w_{BPF}^{(i)}$. After resampling, many resampled bootstrap particles $\tilde{\boldsymbol{x}}_1^{(i),BPF}$ will coincide and all particles will have equal weight $1/N$.

For the APF resampling of the initial state is performed

$$\boldsymbol{x}_0 \approx \sum_{i=1}^{n} w_{APF}^{(i)}\ \delta(\boldsymbol{x}_0 - \boldsymbol{x}_0^{(i)}) \tag{3.44}$$

where

$$w_{APF}^{(i)} \propto n(\|\boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}\|;\ 0, R) \tag{3.45}$$

The effective number of particles is calculated as

$$N_{eff}^{APF} = \frac{1}{\sum_{i=1}^{N} (w_{APF}^{(i)})^2} \tag{3.46}$$

and $\boldsymbol{x}_1^{(i),APF} = \tilde{\boldsymbol{x}}_0^{(i)} + \boldsymbol{\nu}^{(i)}$ with associated weight $1/N$ where $\tilde{\boldsymbol{x}}_0^{(i)}$ are resampled particles. Notice that, because of $\boldsymbol{\nu}^{(i)}$, all particles $\boldsymbol{x}_1^{(i),APF}$ are distinct from one another.

For A1 and A3 (jointly denoted as AN) we use the weights described in Eq. (3.12) and Eq. (3.36), and we calculate the effective number of particles as

$$N_{eff}^{AN} = \frac{1}{\sum_{i=1}^{N} \left(\omega_{AN}^{(i)}\right)^2} \tag{3.47}$$

After sampling from the GMM, all sampled particles $\boldsymbol{x}_1^{(i),AN}$ are distinct from one another and have weight $1/N$.

Performing 100 random experiments for each of the four filters, we obtain the average number of effective particles, the root mean square error (RMSE), and the Cramer-Rao lower bound (CRLB) [80, 102] values given in Table 3.1. The results show that starting from the same initial 300 particles the proposed methodologies produce the most sample diversity and best accuracy among the filters. Notice that none of the algorithms in Table 3.1 (new or existing) approach the Cramer-Rao lower bound, this is true for the following examples as well. This is due to the complex nonlinear nature of the examples chosen.

Table 3.1: Results of Example 1. Number of effective particles

Ex. 1	Effective Particles	RMSE
BPF (300)	9.8370	1.8215
APF (300)	12.4878	1.7847
A1 (300)	56.3863	1.6333
A3 (300)	62.0461	1.6228
CRLB	— —	0.1999

3.3.2 Univariate Nonstationary Growth Model

Consider the discrete time highly nonlinear scalar dynamic system and measurement model given by [30, 51, 52]:

$$x_k = \frac{1}{2}x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos\big(1.2(k-1)\big) + \nu_{k-1} \tag{3.48}$$

$$y_k = \frac{x_k^2}{20} + \eta_k \tag{3.49}$$

where the process noise, ν_{k-1}, and the measurement noise, η_k, are assumed to be independent zero mean Gaussian random variables with variances $Q = 1$ and $R = 1$, respectively.

This model is highly nonlinear and bimodal. The cosine term in the dynamic equation varies with time k. The likelihood has a bimodal nature which makes the states more difficult to estimate. In this example, a Monte Carlo analysis is performed with 200 simulations, each simulation has a time span $k = [0,\ 50]$. The estimation performance of the EKF, UKF, BPF and the three algorithms proposed here [A1, Algorithm II (A2), and A3] are compared based on RMSE, effective sample size (ESS), and noncredibility index (NCI) [55]. The RMSE for each Monte Carlo simulation is calculated from the true and estimated states at each time k. The ESS is the effective number of particles calculated as in the previous example. The NCI is defined as

$$\begin{aligned}\text{NCI}_k = \frac{1}{M}\sum_{j=1}^{M}\big[10\log_{10}\left((\boldsymbol{x}_k^j - \boldsymbol{\mu}_k^j)^{\mathrm{T}}(\boldsymbol{P}_k^j)^{-1}(\boldsymbol{x}_k^j - \boldsymbol{\mu}_k^j)\right) \\ - 10\log_{10}\left((\boldsymbol{x}_k^j - \boldsymbol{\mu}_k^j)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_k^j - \boldsymbol{\mu}_k^j)\right)\big]\end{aligned} \tag{3.50}$$

where M is the number of Monte Carlo simulations, $\boldsymbol{x}_k^j$ are the true states, $\boldsymbol{\mu}_k^j$ are the estimated states, $\boldsymbol{P}_k^j$ are the filter's error covariance matrix of the j-th Monte Carlo run computed with Eq. (2.36), and $\boldsymbol{\Sigma}_k$ is the ensemble error covariance matrix of the estimates at time k computed from the Monte Carlo samples. The NCI quantifies the difference between the ideal error covariance matrix $\boldsymbol{\Sigma}_k$ and the estimated error covariance matrix $\boldsymbol{P}_k$. The NCI metric is a geometric average of 10 times the logarithm of the normalized estimation error squared (NEES) ratio; it is a balanced

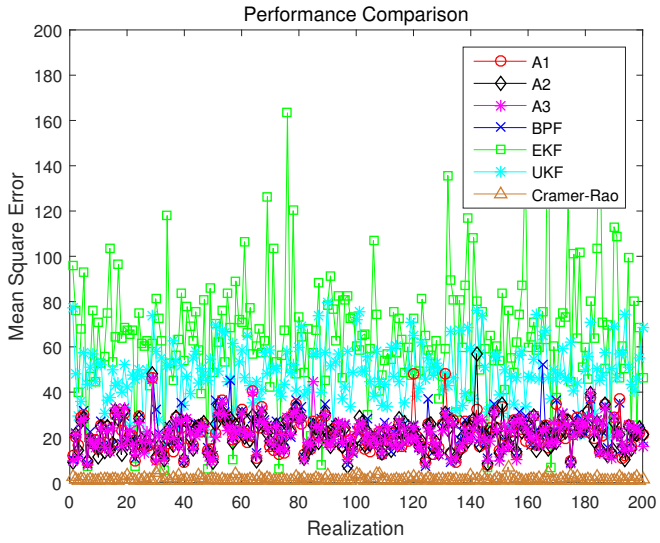


Figure 3.1: Time averaged RMSE for 200 random realizations

measure of the consistency of the estimators. When the difference between $\boldsymbol{\Sigma}_k$ and $\boldsymbol{P}_k$ is small, the NCI value should be zero or nearly zero at all times [55].

The Root Mean Square of the RMSEs, the Monte Carlo averaged ESS, and the NCI from the 200 Monte Carlo runs are shown in Table 3.2. A total of 100 particles are used in both the BPF and the new algorithms proposed here.

Figure 3.1 shows the RMSE and the CRLB of the 200 simulation, the RMSE for each is calculated over a time span of $[0,\ 50]$. The RMSE values of each filter are listed in Table 3.2. Our three proposed algorithms have comparable RMSEs. The best performance is obtained with A3, which starts each iteration from a GMM with non-zero covariance. The RMSEs of the EKF and UKF are higher than that of any sample-based filters. Moreover, the proposed algorithms have better performance than the BPF given the same number of particles, 100.

Table 3.2: Results of Example 2. RMSE for 200 Monte Carlo simulations

Ex. 2	RMSE	ESS	NCI
A1 (100)	22.6558	78.0530	5.3895
A2 (100)	22.4748	77.6704	4.1806
A3 (100)	22.3328	79.0343	4.1823
BPF (100)	23.5081	60.2690	6.0290
EKF	71.7314	— —	17.2206
UKF	50.2221	— —	10.3616
CRLB	1.7258	— —	— —

The consistency test result of each estimator represented by the absolute NCI value is depicted in Figure 3.2 In this figure, the NCI values of our proposed algorithms are smaller than those of other estimators. Figure 3.3 describes the ESS which indicates sample diversity of particle filters. In the figure, the Monte Carlo simulations show that the proposed methodologies produce significantly higher effective number of particles than the BPF. The proposed A3 method performs best in terms of RMSE and ESS.

3.3.3 Lorenz96 system

In this example, the BPF and the here proposed A1 and A3 are applied to a Lorenz96 system [74, 83]. The Lorenz96 dynamical system is expressed as follows:

$$\dot{x}_i(t) = x_{i-1}(t)\left(x_{i+1}(t) - x_{i-2}(t)\right) - x_i(t) + F + \nu_i(t) \tag{3.51}$$

$$\boldsymbol{y}_k = H\ \boldsymbol{X}(t_k) + \boldsymbol{\eta}_k, \quad H_{i,j} = \begin{cases} 1, & j = 2i-1 \\ 0, & \text{otherwise} \end{cases},$$

$$\text{for } i = 1, \cdots, 20,\ j = 1, \cdots, 40 \tag{3.52}$$

where $x_i(t)$, $i = 1, 2, \cdots, 40$, are the components of the 40th-dimensional vector $\boldsymbol{X}(t)$. In the dynamics equation the following conventions are used $x_{-1} = x_{N-1}$,

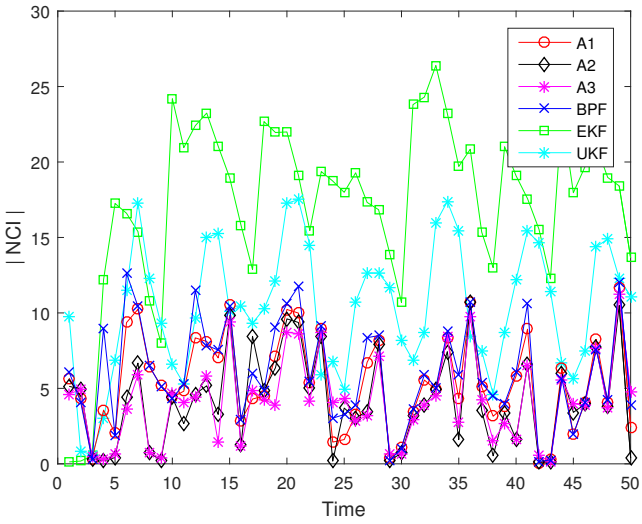


Figure 3.2: Time history of the absolute NCI value for 200 random realizations

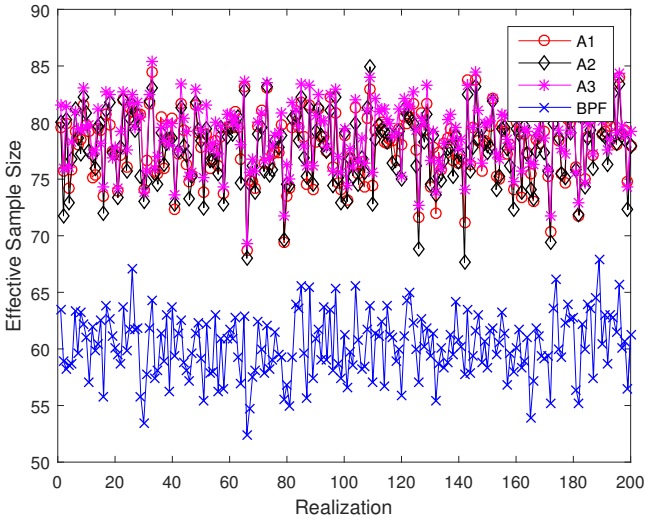


Figure 3.3: Time averaged effective sample size for 200 random realizations

$x_0 = x_N$, and $x_1 = x_{N+1}$. The term F represents a constant external forcing and is set to 8, which causes chaotic behavior in the system. The dynamics is propagated for 10 seconds at 20 Hz while the discrete measurements are available at 1 Hz, $t_k = 1, 2, \cdots, 200$. Fourth order Runge-Kutta integration is used with a step size of 0.05 sec, and the process noise is held constant over each 0.05 second interval with zero correlation between the intervals. The measurements are linear and measure only the components of the state vector that have odd indices. It is assumed that the process noise and measurement noise are uncorrelated, white, zero mean, and with covariance matrices given by $Q = 10^{-2}$ and $R = 10^{-2}\boldsymbol{I}_{20\times 20}$, respectively [83]. The initial state of the system is assumed multivariate Gaussian distribution with $\boldsymbol{\mu}_0 = F[1, 1, \cdots, 1]^{\mathrm{T}}$ and $P_0 = 10^{-3}\boldsymbol{I}_{40\times 40}$.

Figure 3.4 shows the CRLB and the performance of 100 Monte Carlo simulations with 2000 particles for the Lorenz96 system. The time averaged value of RMSE of the three algorithms and the CRLB are shown in Table 3.3. For such a large system, A2 is not recommended because of high computation time and is omitted from this example. The results show that the performance of A3 is better than the A1. Moreover, the BPF is found to provide significantly inferior performance. To compare the consistency of the filters, the absolute NCI value is computed and compared in Figure 3.5. This figure indicates that the performance of A1 and A3 are comparable. On the other hand, the absolute NCI value of the BPF is greater than that of A1 and A3 over time. The time averaged ESS for the 100 Monte Carlo simulations are shown in Figure 3.6. The effective number of particles for the BPF is small since it does not directly account for the latest information of

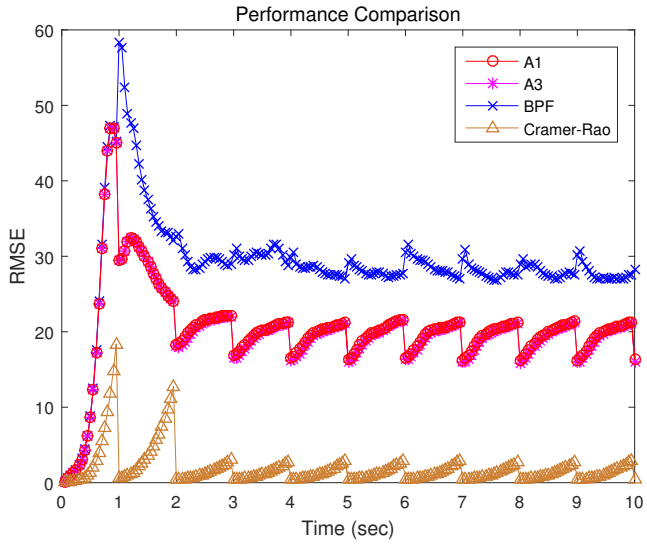


Figure 3.4: Monte Carlo averaged RMSE for 100 random realizations

the measurement. On the other hand, A1 and A3 provide good diversity and number of effective particles. The quantitative results representing the consistency and ESS of the filters are listed in Table 3.3. A3 has the best performance in terms of accuracy, consistency, and ESS.

Table 3.3: Results of Example 3. RMSE for 100 Monte Carlo simulations

Ex. 3	RMSE	ESS	NCI
A1 (2000)	20.3819	1248.5274	0.4160
A3 (2000)	20.1415	1338.7312	0.3920
BPF (2000)	28.7860	229.0955	22.7186
CRLB	1.8377	— —	— —

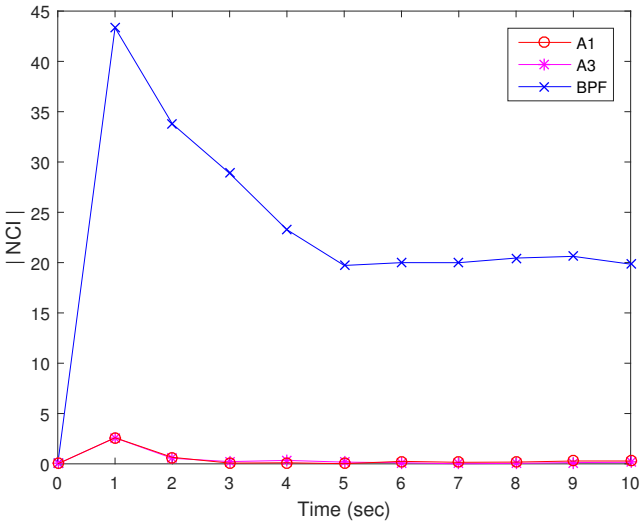


Figure 3.5: Time history of the absolute NCI value for 100 random realizations

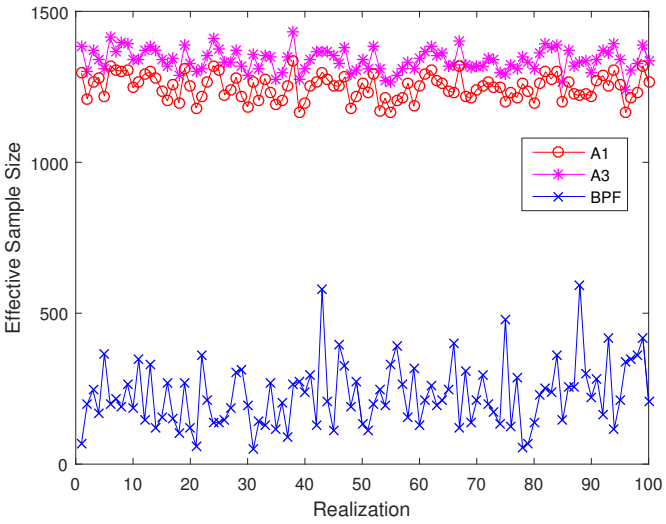


Figure 3.6: Time averaged effective sample size for 100 random realizations

3.3.4 The Blind Tricyclist Problem

In this last example, A3 is tested on the blind tricyclist problem presented in Ref. [80], and its performance is compared to that of the EKF and RPF. The blind tricyclist is a challenging nonlinear estimation problem with seven states consisting of unknown planar position, heading angle, and four observation parameters. Unlike the previous examples, in this problem the process noise does not enter the dynamics linearly. Moreover, the process noise covariance matrix is not full rank because three states do not have process noise. Therefore, most particle filters will fail to produce particle diversity, while A3 and the RPF are suitable and applied to this problem. The dynamics are propagated for 141 seconds at 2 Hz with the two known inputs corrupted by additive Gaussian noise. Two relative bearing measurements are available every 3 seconds out-of-phase at $180\,^{\circ}$, e.g., the rider gets relative bearing measurements from two shouting friends: the first friend shouts out at sample times 0.5, 3.5, 6,5, etc., while the second friend shouts out at sample times 2, 5, 8, etc.

Figure 3.7 displays the time history of the position's RMSE magnitude of the CRLB and 100 Monte Carlo simulations of the EKF, A3 with 3000 and 10000 particles, and RPF with 3000 and 10000 particles. Since the process noise is only related to the planar position and heading states, the process noise covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ is not full rank. Therefore, A1 and A2 cannot be successfully applied to this problem, neither are the BPF and APF. In addition, since the tricycle heading angle and the merry-go-round phase angles can cause a 2π cycle ambiguity, a 2π relative unwrapping operation is performed. The RPF resampling is done

whenever the number of effective particles is smaller than a resampling threshold $\hat{N}_{eff}$, chosen as 400 and 5000 for 3000 and 10000 particles, respectively [81]. The results indicate that the performances of both the A3s with 3000 and 10000 mixture elements are better than those of the RPFs for the first 100 secs but they become comparable after that. The reason is that the RPF implemented here uses an Epanechnikov kernel density estimator, which is optimal for Gaussian distributions. After 100 seconds of simulation time, when the total uncertainty of the problem reduces, the distribution looks "more" Gaussian and the RPF performs really well. However, when the PDF differs substantially from Gaussian, the Epanechnikov kernel density estimator and hence the RPF perform noticeably worse than A3. If the posterior density was known, an optimal kernel estimator could be found to produce excellent results. Generally speaking, however, the shape of the posterior distribution is unknown and thus A3 does a better job of representing the distribution, as the consistency test below clearly shows. Ref. [80] details the reason why RPF with 10000 particles performs worse than the RPF with 3000 particles: "First, the increase from 3000 to 10,000 particles might be insufficient to ensure improvement in a 100-run Monte Carlo simulation. Also, the RPF regularization's dithering might have interfered with the PF's accuracy convergence in the limit of a large number of particles."

The RMSE of A3 and RPF lie a bit lower than the CRLB during the first 5 sec of the run, which is theoretically impossible but allowable since a finite number of Monte Carlo simulations is conducted [82]. This figure also shows that the performance of the EKF is inferior to that of A3 and the RPF. The quantitative RMSE

results for position is listed in Table 3.4. The RMSE value of the A3 with 3000 particles is 3.97% smaller than that of the RPF with 3000 particles and the RMSE value of the A3 with 10000 particles is 27.87% smaller than that of the RPF with 10000 particles.

Figure 3.8 shows the absoulte NCI value of each estimator. The absolute value of NCI of all filters increases as time passes. This is because the process noise covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ is rank-deficient. It is well known that small process noise can causes degeneracy in particle filters, thus degrading their performance [7]. The figure shows that the RPF with 3000 particles does suffer from degeneracy. Even with 3000 particles, the absolute value of NCI of A3 shows that the filter is performing in a very satisfactory fashon. The time averaged absolute NCI value to the total samples of 100 cases is listed in Table 3.4, where n/a indicates degeneracy. The average computation time per filtering run in MATLAB on a 3.5-GHz, four-core Ubuntu operation system is also presented in Table 3.4. The absolute NCI value of the A3 is smaller than that of the EKF and RPF with the same number of particles. In addition, compared to the RPF, the A3 reduces the mean computation time by 5.65% and 19.72% with 3000 and 10000 particles, respectively. Therefore, the performance in terms of accuracy, consistency, and mean computation time of the proposed algorithm is conspicuously better than that of the EKF and RPF.

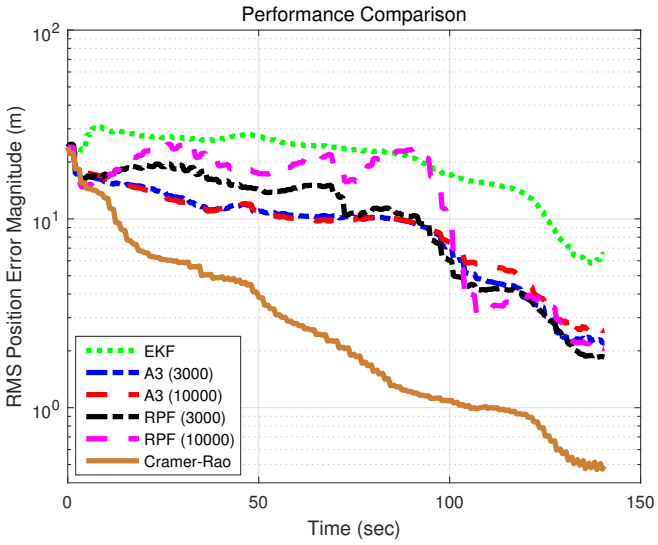


Figure 3.7: Monte Carlo averaged RMSE for 100 random realizations

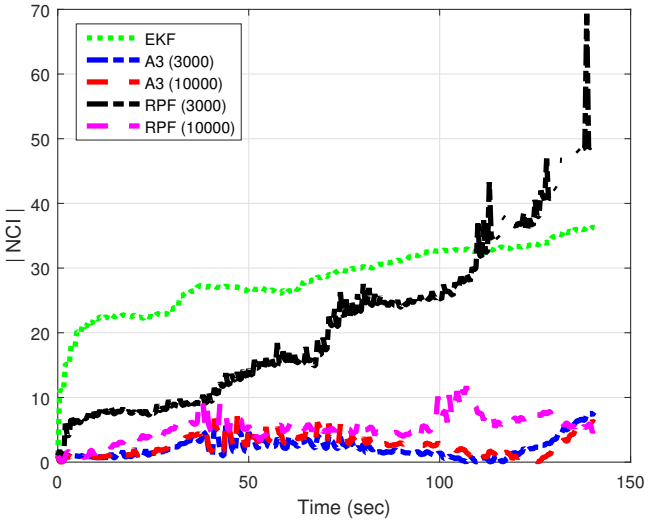


Figure 3.8: Time history of the absolute NCI value for 100 random realizations

Table 3.4: Results of Example 4: RMSE for 100 Monte Carlo Simulations

Ex. 4	**RMSE**	**NCI**	**Computation time (sec/sim.)**
EKF	21.2319	28.3933	0.0205
A3 (3000)	11.0344	2.1958	41.3793
A3 (10000)	10.8863	2.6238	170.2232
RPF (3000)	11.4904	n/a	43.8562
RPF (10000)	15.0916	5.3220	212.0285
CRLB	3.9161	— —	— —

3.4 Chapter's Summary

In this chapter, a new sequential Monte Carlo algorithm is proposed that samples from a Gaussian mixture model approximation of the posterior distribution. Each sample of the distribution at the prior time is treated as a Gaussian component with a collapsed zero covariance matrix. Process noise is responsible for generating propagated components with non-singular covariance matrix, and the Gaussian sum filter algorithm is used to calculate the posterior distribution. It is shown that the proposed algorithm improves over the accuracy, consistency, and effective number of particles of the bootstrap and regularized particle filters in the numerical examples considered.

Two small modifications of the baseline algorithm are also proposed to further improve its accuracy. First, an importance sampling version of the algorithm is developed. At the cost of more computations, this modified approach slightly improves over the baseline algorithm. In the second modification, the initial covariance of the Gaussian mixture model components is not set to zero, but to a small value that removes the bias in the sample covariance, this approach is necessary, for

example, when the process noise is not sufficient to produce nonsingular covariance matrices for the components. All the proposed algorithms have better performance than the conventional bootstrap particle filter in all tests performed.

Chapter 4

Kernel-Based Ensemble Gaussian Mixture Filtering for Orbit Determination with Sparse Data

4.1 Motivation and Comparison with State of the Art

The algorithms introduced in the previous chapter were shown to improve the performance of the BPF and RPF under challenging nonlinear scenarios. The process of making each particle a Gaussian component improves particle diversity and overall performance of the filter. The three proposed algorithms, however, suffer from one common drawback of most particle filters: poor performance in the absence of process noise. This is particularly true for little to no process noise during long propagation times in-between measurements. If an initial uncertainty grows considerably due to the dynamics, a set of particles that is adequate to approximate the initial PDF might become too sparse and be inadequate to represent the *a priori* PDF for filtering purposes. This chapter investigates this problem in the context of orbit determination.

In recent years, there has been an increasing interest in tracking an ever-growing number of SOs for collision avoidance and space domain awareness [16, 20, 77]. As very large, low earth orbit (LEO) constellations are being developed and launched, the risk of collision in LEO keeps increasing because of a high density of SOs in this region. The high number and density of LEO SOs require accurate orbit determination and data association [71]. Currently, only a limited number of radar-based surveillance sensors are available and used to estimate the state of an SO in LEO. The current approach to maintaining a LEO catalogue is not scalable to tens of thousands of spacecrafts. The solution of this problem is either adding more hardware (more tracking stations and/or clusters of supercomputers) or improving the computational efficiency of tracking and data association software used

to maintain the catalogue.

A software-only solution is one in which the number of available measurements per SO is reduced because the current surveillance network is tasked to acquire data from many more SOs. It requires an efficient data association algorithm and an estimator able to extract as much information as possible from the sparse data. This chapter addresses the latter, and proposes an accurate and computationally fast nonlinear estimation algorithm for orbit determination.

For linear systems with linear measurements, the well-known Kalman filter [49] provides a globally optimal solution, i.e., it extracts as much information from the data as possible (in a MMSE sense). In the presence of nonlinearities (either in the dynamics, the measurements, or both), a nonlinear filter is able to produce a more accurate estimate than a linear one, i.e., extract more information from the data. Radar measurements of range, range-rate, and angles to an SO are inherently nonlinear. A nonlinear filter, therefore, will outperform a linear filter such as the EKF [29] or UKF [47] even in the presence of near-linear dynamics.

To cope with the sparse data problem, this chapter compares two nonlinear algorithms: the adaptive entropy-based Gaussian mixture information synthesis (AEGIS) [19] and the kernel-based ensemble Gaussian mixture filtering (EnGMF) [3]. This work proposes a new modification to EnGMF to greatly improve its computational complexity. Two implementations of the UKF are also compared to the proposed approach, representing the SO with both Cartesian and equinoctial coordinates [14].

For linear measurements and dynamics, the UKF reduces to the Kalman filter and is the optimal output of all linear estimators regardless of the probability distributions (in a MMSE sense). The UKF is typically more robust to nonlinearities than the EKF [47] but can still fail to produce an adequate estimate in the case of high nonlinearities. The nonlinearities of orbital dynamics are easily mitigated by choosing to represent the SO's state with an appropriate set of orbital elements, for example equinoctial elements. Changes in these elements, specifically the angle quantity, are linear, and variations due to nonlinear effects are relatively small. This choice of coordinates, therefore, allows for accurate and computationally inexpensive long time propagations of the mean and covariance matrix [48, 88], for example when using the unscented transformation (UT). The price to pay for linear dynamics is typically an even more nonlinear measurement model, which may cause UKF divergence in a scarce-measurement environment, as shown in the numerical results section of this chapter. In measurement-rich environments, when long propagations are followed by dense measurements arcs, a batch least-squares approach is often the preferred orbit determination solution [98], as it allows to extract more information from nonlinear measurements than linear sequential filters. After processing the measurement batch, the mean and covariance of the estimate can be propagated with the UT to start a new iteration. Batch least squares does not provide full information about the probability distribution function, and it only returns the mean and covariance matrix and the underlying distribution is typically assumed Gaussian, hence they work best when many measurements are available such that the resulting uncertainty is close to Gaussian. Nonlinear recursive filters,

on the other hand, approximate the optimal MMSE estimator, which has the lowest square estimation error (on average), and provides a full description of the underlying PDFs. AEGIS and our proposed modification to EnGMF are two examples of nonlinear filters.

While this chapter focuses on the estimation problem, adding data association to a single-target nonlinear filter is a well-studied problem. Data association and collision detection benefit from full knowledge of the PDF, which can be approximated with AEGIS and the EnGMF but inevitably results in a Gaussian assumption for linear estimators. As long as the PDF remains approximately Gaussian after measurements are incorporated, linear filters produce excellent performance. Ref. [41], for example, assumes that the initial orbit determination solution is an estimate with a Gaussian distribution, and employs modified equinoctial elements to propagate the state and associate a sequence of observations to an SO using the Mahalanobis distance.

The AEGIS method is based on the standard GSF [2, 97]. The GSF is a nonlinear estimator for nonlinear systems and it has been applied to SO tracking applications [38, 39]. To deal with multimodal and non-Gaussian distributions, the GSF approximates the PDF as a GMM. The GSF provides a nearly optimal solution when enough components are taken and each Gaussian component has a small enough covariance matrix such that the nonlinear dynamic and measurement functions can be accurately approximated to linear functions in the support of each Gaussian component. In the presence of a Gaussian prior and a nonlinear measurement, the GSF outperforms linear filters when the prior is approximated by many

Gaussian of smaller covariance such that the measurement is approximately linear in their support.

One of the limitations of the standard GSF is that the weights of the Gaussian components remain the same during nonlinear propagations. Several studies recently have been proposed to address this issue and improve the standard GSF algorithm to better account for nonlinear dynamics [19, 101, 103]. One of these approaches is AEGIS, which splits the Gaussian components to reduce the effects of nonlinearities of a dynamical system during the prediction of state uncertainty [19].

Another approach to nonlinear filtering is SISR, commonly known as particle filters (PFs) [8]. PFs are known to suffer from degeneracy with near-deterministic dynamics, i.e., with little process noise. As orbital dynamics is well characterized, a particle filter implementation of orbit determination inevitably requires low process noise. Modifications have been investigated to improve the standard SISR methods such as the BPF, APF, and RPF [23] by combining particle filters and GSF [3, 64, 84, 112, 113]. For example, the sequential Monte Carlo filtering with Gaussian mixture model (SMCGMM) proposed in Chapter 3 [112] assumes that each particle of the pre-propagation distribution to be a Gaussian component having a zero or small covariance matrix. Refs. [84] and [113] integrate a PF with a clustering algorithm (e.g., K-means algorithm or EM algorithm) to approximate the prior distribution with a GMM. Although clustering to form the GMM provides an accurate solution for a highly nonlinear system, it is computationally expensive and not of practical use for tracking large LEO constellations.

Other examples of hybrid PF/GSF algorithms include Refs. [3] and [64]

which approximate each propagated particle as a Gaussian component with a non-zero covariance matrix calculated by bandwidth selection for kernel density estimation (KDE). KDE is a non-parametric technique to estimate the PDF of a random variable [93]. The KDE algorithm with a Gaussian kernel is similar to the EM clustering algorithm in that they construct a GMM using the particles. However, in the KDE algorithm, every particle is considered as a Gaussian component to establish a GMM whereas EM clustering algorithm groups several particles into each Gaussian mixture components. An adaptable bandwidth selection suffers from a high computational cost similar to the clustering algorithms presented in [84] and [113].

In this chapter, the EnGMF algorithm is modified to efficiently track SOs in LEO with short and sparse observation data. A key element of the EnGMF algorithm is the determination of the covariance matrix of each Gaussian component in a GMM. The covariance matrix is determined by the bandwidth parameter of a kernel function. Although the optimal bandwidth parameter can be obtained using a data-driven method [44, 76], this approach is computationally expensive. Alternatively, we can compare the simulation results of a system using a range of the bandwidth parameter [64] and tune this parameter according to the system. In this chapter, we propose an approach to achieve a near-optimal bandwidth parameter with a low computational cost for orbit determination with sparse observation data. We achieve this by computing the bandwidth of a Gaussian kernel in the KDE algorithm with Silverman's rule of thumb [94] to reduce the KDE computational burden.

The remainder of this chapter is organized as follows. First, the dynamics

and measurement models are described and the coordinate systems are presented. Then, the two nonlinear estimation techniques, the AEGIS and a modified EnGMF, are introduced in Section 4.3. In Section 4.4, simulation results are shown using the proposed algorithms followed by some concluding remarks on the methodology and results.

4.2 System Models

This section serves to introduce system models used in this chapter.

4.2.1 Dynamics Model

The inertial position and velocity of an SO are denoted by $\boldsymbol{r}^I = [x\ y\ z]^{\mathrm{T}}$ and $\boldsymbol{v}^I = [v_x\ v_y\ v_z]^{\mathrm{T}}$. The orbital dynamics of an SO in Earth-Centered Inertial (ECI) coordinates are given by

$$\begin{bmatrix} \dot{\boldsymbol{r}}^I \\ \dot{\boldsymbol{v}}^I \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^I \\ -\frac{\mu}{r^3}\boldsymbol{r}^I + \boldsymbol{a}^I_{NS} + \boldsymbol{a}^I_{3B} + \boldsymbol{a}^I_{drag} + \boldsymbol{a}^I_{srp} \end{bmatrix} \tag{4.1}$$

where μ is the Earth's gravitational parameter and r is the Euclidean norm of $\boldsymbol{r}^I$. $\boldsymbol{a}^I_{NS}$ is the gravitational perturbation due to non-spherical effect of the Earth gravity, $\boldsymbol{a}^I_{3B}$ indicates the third-body perturbations of the Moon and the Sun, and $\boldsymbol{a}^I_{drag}$ and $\boldsymbol{a}^I_{srp}$ represent the acceleration perturbation due to atmospheric drag and solar radiation pressure (SRP), respectively. For this study, the EGM2008 [78] gravity model is used for the Earth and 70 $\times$ 70 degrees and order are applied for gravity modeling, and the planetary and lunar ephemeris DE430 [27] is selected to compute the location of the Moon and the Sun.

The primary non-gravitational force acting on SOs in LEO is the drag force. The drag acceleration due to atmospheric density relies upon the drag coefficient, the cross-sectional area of an SO face perpendicular to velocity vector, and the height of an SO above the Earth's surface. The acceleration due to drag is then given by

$$\boldsymbol{a}^I_{drag} = -\frac{1}{2} C_d \frac{A}{m} \rho_d \|\boldsymbol{v}_{rel}\| \boldsymbol{v}_{rel} \tag{4.2}$$

where C_d is the drag coefficient, m and A are the mass and cross-sectional area of the SO, respectively, ρ_d is the atmospheric density at altitude of the SO, $\|\cdot\|$ means the Euclidean norm, and $\boldsymbol{v}_{rel}$ is the atmosphere-relative velocity vector. For computing the atmospheric density, the exponential density model is employed in this study [105].

The acceleration due to SRP depends on the shape of an SO and the cannonball model, i.e., spherical object, is assumed in this chapter. The acceleration perturbation due to SRP is then given by

$$\boldsymbol{a}^I_{srp} = -\frac{S_F A C_R \boldsymbol{u}^I_{sun}}{mc} \tag{4.3}$$

where S_F is the solar flux, m is the mass of the SO, c is the speed of light, C_R is the coefficient of reflectivity, u^I_{sun} is the unit vector pointing from the SO to the Sun in the ECI frame.

4.2.2 Measurement Model

Range ρ_{range} and range-rate $\rho_{rangerate}$ measurements along the line of sight (LOS) from a ground-based radar sensor to an SO is provided. The relative position

vector $\boldsymbol{\rho}^I = [\rho_x\ \rho_y\ \rho_z]^{\mathrm{T}}$ between the SO and a ground station $\boldsymbol{r}_S^I$ coordinatized in ECI is given by:

$$\boldsymbol{\rho}^I = \boldsymbol{r}^I - \boldsymbol{r}_S^I \tag{4.4}$$

The error-free range measurement is given:

$$\rho_{range} = \|\boldsymbol{\rho}^I\| = \|\boldsymbol{r}^I - \boldsymbol{r}_S^I\| \tag{4.5}$$

By differentiating Eq. (4.5) with respect to time, the error-free range-rate measurement is obtained as follows:

$$\rho_{rangerate} = \frac{\left(\boldsymbol{r}^I - \boldsymbol{r}_S^I\right)^{\mathrm{T}}\left(\boldsymbol{v}^I - \boldsymbol{v}_S^I\right)}{\|\boldsymbol{r}^I - \boldsymbol{r}_S^I\|} \tag{4.6}$$

where $\boldsymbol{v}_S^I$ is the time rate of change of the ground station position vector with respect to the inertial frame.

Along with the range and range-rate, angle data in the form of right ascension α and declination δ are measured to estimate the states of the SO. The error-free angle observation equations are described as follows:

$$\alpha = \tan^{-1}\left(\frac{\rho_y}{\rho_x}\right), \quad \delta = \sin^{-1}\left(\frac{\rho_z}{\|\boldsymbol{\rho}^I\|}\right) \tag{4.7}$$

All measurements are corrupted by zero-mean, Gaussian noise. In this study, light travel time delay and measurement biases are not considered.

4.2.3 Coordinate systems

The dynamic equations of the SO presented above are expressed in Cartesian coordinates, which results in nonlinear differential equations. Alternatively,

equinoctial orbital elements [14] offer a near-linear dynamics. The Keplerian motion is exactly linear, and nonlinearities arise only due to perturbations such as non-central gravity and drag. The equinoctial orbital elements are expressed as functions of the Keplerian orbital elements as follows:

$$\begin{aligned} a &= a \\ h &= e \sin(\omega + \Omega) \\ k &= e \cos(\omega + \Omega) \\ \lambda_0 &= M_0 + \omega + \Omega \\ p &= \tan(i/2) \sin(\Omega) \\ q &= \tan(i/2) \cos(\Omega) \end{aligned} \tag{4.8}$$

where a is the semi-major axis, e is the eccentricity, i is the inclination, Ω is the longitude of the ascending node, ω is the argument of periapsis, and M_0 is the mean anomaly.

4.3 Estimation Techniques

This section reviews the AEGIS and introduces the proposed nonlinear estimation algorithms to cope with the sparse data problem: a modified EnGMF.

4.3.1 Adaptive Entropy-based Gaussian Mixture Information Synthesis

The AEGIS uses an entropy-based method to detect nonlinearity of a dynamical system during the prediction of state uncertainty and then applies a splitting technique to decrease the approximation error caused by truncating the nonlinear functions of the system to low-order. The AEGIS method is based on the standard

GSF which is a nonlinear estimator. In the GSF, non-Gaussian PDFs are approximated as a GMM as follows:

$$p(\boldsymbol{x}) = \sum_{i=1}^{N} \omega^{(i)} n(\boldsymbol{x}; \boldsymbol{\mu}^{(i)}, P^{(i)}) \tag{4.9}$$

where $\boldsymbol{x}$ is a random variable, $p(\boldsymbol{x})$ is the PDF of $\boldsymbol{x}$, N is the number of all Gaussian components, $n(\boldsymbol{x}|\boldsymbol{\mu}, P)$ represents the Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance P; and $\boldsymbol{\mu}^{(i)}$, $P^{(i)}$, and $\omega^{(i)}$ are the means, covariance matrices, and weights of the i^{th} Gaussian component. The PDF normalization and positivity properties lead to the following constraints on the weights:

$$\omega^{(i)} \geq 0, \ \forall i \qquad \sum_{i=1}^{N} \omega^{(i)} = 1 \tag{4.10}$$

The performance of the GSF mainly depends on both the number and the weights of the components of a GMM; however, both of them are held constant during the propagation step. To improve the standard GSF algorithm to better adapt to nonlinearities of the system, the AEGIS approach allows for the modification of the Gaussian components over the propagation step based on two main mechanisms.

The first step of the AEGIS is to monitor the nonlinearity of the dynamics using a property derived from the differential entropy for linearized dynamical systems. The differential entropy of a continuous random variable $\boldsymbol{x}$ is defined as follows [19]:

$$H(\boldsymbol{x}) = -\int_S p(\boldsymbol{x}) \log\left(p(\boldsymbol{x})\right) d\boldsymbol{x} = E\{-\log\left(p(\boldsymbol{x})\right)\} \tag{4.11}$$

where S is the support set. In this chapter, all logarithms are assumed to be natural. The analytic solution of the differential entropy for a multivariate Gaussian

distribution is then expressed as follows:

$$H(\boldsymbol{x}) = \frac{1}{2}\log|2\pi e P| \tag{4.12}$$

where P is the covariance matrix and $|\cdot|$ represents the matrix determinant. By taking a derivative with respect to time for Eq. (4.12), the time rate of the differential entropy can be calculated as follows:

$$\dot{H}(\boldsymbol{x}) = \frac{1}{2}\text{trace}\{P^{-1}\dot{P}\} \tag{4.13}$$

where $\dot{P}$ is the time rate of change of the covariance matrix, which in the absence of process noise evolves as:

$$\dot{P}(t) = F\left(\boldsymbol{\mu}(t), t\right) P^{\mathrm{T}}(t) + P(t) F^{\mathrm{T}}\left(\boldsymbol{\mu}(t), t\right) \tag{4.14}$$

where $\boldsymbol{\mu}(t)$ is the time-varying mean of the Gaussian distribution and $F\left(\boldsymbol{\mu}(t), t\right)$ is the Jacobian of the dynamics evaluated at the mean $\boldsymbol{\mu}(t)$. By substituting Eq. (4.14) into Eq. (4.13), the time rate of the differential entropy for a linearized dynamical system is obtained as follows:

$$\dot{H}(\boldsymbol{x}) = \text{trace}\{F\left(\boldsymbol{\mu}(t), t\right)\} \tag{4.15}$$

The entropy value for a linearized system, therefore, can be calculated by numerically integrating Eq. (4.15) with an appropriate initial condition, which requires only the evaluation of the trace of the dynamics Jacobian. On the other hand, a nonlinear determination of the differential entropy can be evaluated via Eq. (4.12) by a nonlinear implementation of the integration of the covariance matrix; for example, unscented transformation is one of the most popular and effective methods

Table 4.1: Three-component splitting library

i	ω_i	μ_i	σ_i
1	0.2252246249	-1.0575154615	0.6715662887
2	0.5495507502	0	0.6715662887
3	0.2252246249	1.0575154615	0.6715662887

for moment evaluation. Any deviation between the linear and nonlinear values of the entropy then indicates that nonlinearity is impacting the Gaussian component. As a result, the difference between the linearized and nonlinear predictions of the entropy can be monitored without the full solution to both the linearized and nonlinear predictors. In other words, when the difference between these values of entropy exceeds a preassigned threshold, a splitting algorithm is applied to the Gaussian component during a propagation. A smaller threshold leads to more frequent splitting during the propagation.

Once the nonlinear effects have been detected from the first step, a splitting algorithm is applied to mitigate the effects by replacing a Gaussian component with several Gaussian components. For the univariate case, each Gaussian component can be decomposed into 3 components using splitting libraries which are shown in Table 4.1. The splitting technique for a univariate case with splitting library can be then extended to the multivariate case by considering the principal directions of the covariance matrix. The details of the algorithm are explained in Ref. [19]. After the propagation, the *a posteriori* mean and covariance matrix, and mixture weights are obtained using the measurement update of the standard GSF.

When allowing the number of Gaussian components to grow unbounded,

the AEGIS is an accurate and consistent estimator. In this chapter, we are interested not only in estimation accuracy, but also in computational efficiency to maintain custody of a very large number of SOs. The proposed solution to achieve this balance of performance versus accuracy is introduced next.

4.3.2 Modified Kernel-Based Ensemble Gaussian Mixture Filtering

As a recursive algorithm, the knowledge of the distribution $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1})$ at the prior time is assumed and approximated by N i.i.d. samples $\boldsymbol{x}_{k-1}^{(i)}$ such that

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{4.16}$$

where $\boldsymbol{y}$ is a measurement vector and $\delta(\cdot)$ is the Dirac delta function. Following the same procedure as the BPF [7], a set of samples at the next time step is obtained using the Markov transition kernel $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. The Markov kernel indicates the dynamics of a system and all estimators use the true dynamic model without process noise in this chapter.

The next step is to convert the samples into Gaussian mixtures using KDE. In other words, each particle is considered as a Gaussian component with non-zero covariance. The approximated GMM of the propagated samples is then expressed as follows:

$$p(\boldsymbol{x}_k) \approx \sum_{i=1}^{N} \frac{1}{N} n(\boldsymbol{x}_k; \boldsymbol{x}_{k|k-1}^{(i)}, B) \tag{4.17}$$

where the bandwidth matrix B is can be calculated by [64]

$$B = \beta \hat{P} \tag{4.18}$$

where β is the bandwidth parameter, $0 \leq \beta \leq 1$, and $\hat{P}$ is the sample covariance matrix calculated from the particles. The Gaussian components' means are the particles $\boldsymbol{x}_{k|k-1}^{(i)}$ and all GMM weights are equal to $1/N$. The covariance matrix of each Gaussian component is determined by the bandwidth parameter. The larger bandwidth parameter β, the smaller the probability assigned to the particle and vice versa.

Finally, we can incorporate the measurement information by updating the means, covariance matrices, and the weights of all N Gaussian components in the same way as the measurement update of the GSF. N i.i.d. samples are then drawn from the GMM approximation of the posterior distribution. These samples are used as a starting point for the next iteration. The details of the measurement update of the GSF and the method to draw N i.i.d. samples from a GMM are explained in Chapter 3.

In the EnGMF algorithm, it is crucial to choose the most appropriate bandwidth which determines the performance of the filter. Bandwith selection is an accuracy vs. computational cost trade off, with the most accurate algorithms numerically solving an optimization problem. In this chapter, we propose to use Silverman's rule of thumb [94] to estimate the bandwidth (i.e., covariance) matrix B_S as follows:

$$B_S = \beta_S \hat{P} = \left(\frac{4}{n_{\boldsymbol{x}} + 2} \right)^{\frac{2}{n_{\boldsymbol{x}}+4}} N^{-\frac{2}{n_{\boldsymbol{x}}+4}} \hat{P} \tag{4.19}$$

We can, therefore, obtain a near-optimal bandwidth parameter for orbit determination with sparse observation data without the need of performing any numerical optimization. If the sampling distribution were Gaussian, Silverman's rule of

thumb would provide the optimal bandwidth parameter based on the mean integrated square error (MISE) as a performance criterion [94]. However, it may result in conservative (large) estimates when the distribution is not close to Gaussian. This is a very desirable feature, since inaccuracies results in conservatism rather than over-confidence and divergence. The flow chart of the modified EnGMF for orbit determination is described in Figure 4.1.

4.4 Numerical Results

To evaluate the performance of the UKF, AEGIS, and EnGMF, one numerical example is considered. The system dynamic equations are numerically integrated with an embedded Runge-Kutta 8(7) method [22]. Range, range-rate, and angle measurements are simulated using a ground station located at the North Pole (latitude = 90°, longitude = 0°, altitude = 0 km), which is an ad-hoc method not to be affected by Earth's rotation. In this simulation, observation data are short and sparse, which means that the observation interval time is much longer than the observation duration. The measurements are available every 10 seconds with a pass lasting only 2 minutes, i.e., 12 measurements per pass. Each observation consists of range, range-rate, right ascension, and declination and the measurements are corrupted by additive zero-mean Gaussian white noise with standard deviation of 30 m and 0.3 m/s for the range and range-rate, respectively, and 100 arc-seconds on the right ascension and declination observation.

The SO is in a near polar orbit with the following Keplerian orbital elements: a = 7,078.0068 km, e = 0.01, i = 85°, and $\omega = \Omega = \nu = 0$. The simulation epoch is

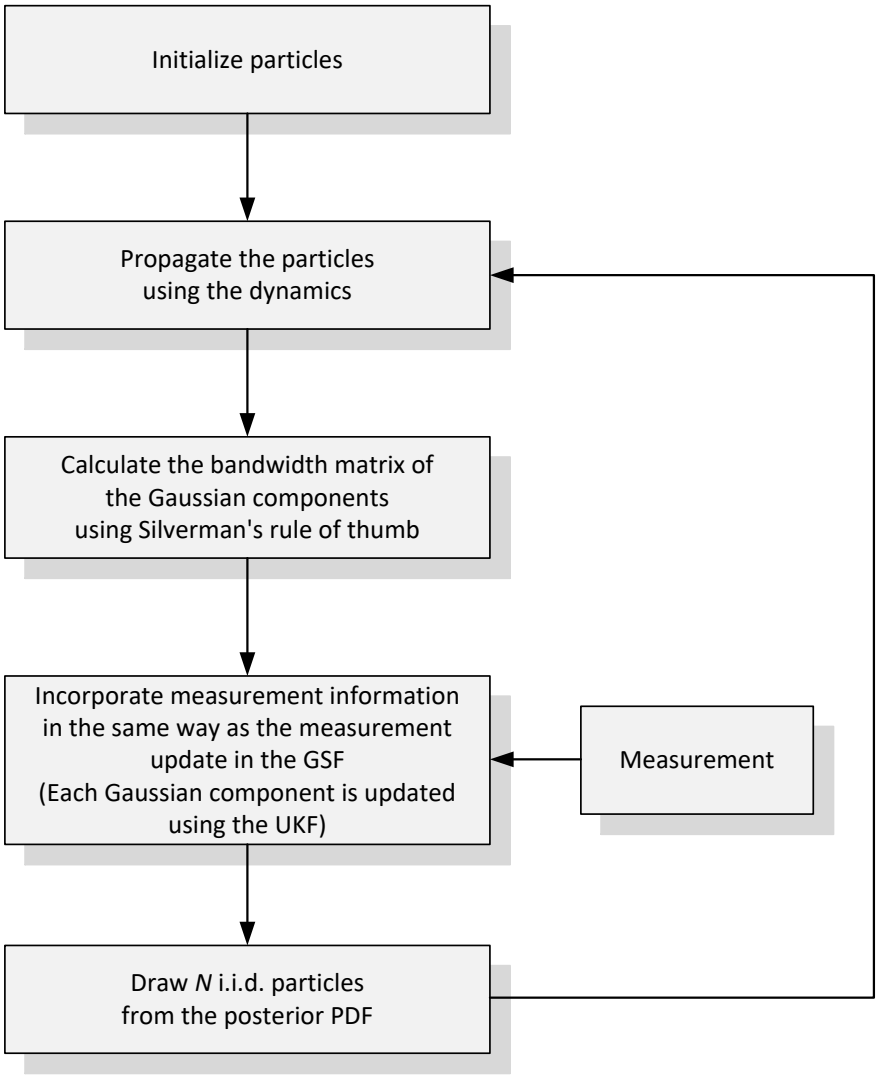


Figure 4.1: The flow chart of the modified EnGMF for orbit determination

4-January-2010 at 00:00:00 UTC. The shape of the SO is assumed to be a sphere with a cross-sectional area of 1 m^2 and a mass of 500 kg. The drag coefficient and the coefficient of reflectivity of the SO are set to be 2 and 1.5, respectively. The

initial distribution is defined in Cartesian coordinates as

$$\boldsymbol{x}_0 \sim n\left(\boldsymbol{x}_0;\ \boldsymbol{\mu}_0,\ P_0\right) \tag{4.20}$$

where

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 7007.2175 & (km) \\ 0 & (km) \\ 0 & (km) \\ 0 & (km/s) \\ 0.6606 & (km/s) \\ 7.5509 & (km/s) \end{bmatrix} \tag{4.21}$$

$$P_0 = \begin{bmatrix} 1.481e\text{+}2 & 0 & 0 & 0 & \text{-}9.237e\text{-}2 & \text{-}5.333e\text{-}2 \\ 0 & 2.885e\text{+}1 & 9.994 & \text{-}3.232e\text{-}2 & 0 & 0 \\ 0 & 9.994 & 5.770 & \text{-}1.242e\text{-}2 & 0 & 0 \\ 0 & \text{-}3.232e\text{-}2 & \text{-}1.242e\text{-}2 & 3.687e\text{-}5 & 0 & 0 \\ \text{-}9.237e\text{-}2 & 0 & 0 & 0 & 6.798e\text{-}5 & 3.145e\text{-}5 \\ \text{-}5.333e\text{-}2 & 0 & 0 & 0 & 3.145e\text{-}5 & 3.166e\text{-}5 \end{bmatrix} \tag{4.22}$$

First, a Monte Carlo analysis is performed with 100 simulations, and each simulation has one measurement pass every orbital period, 5926 seconds. Note that, throughout this paper, the starting time of each measurement pass is randomly selected in close proximity of a multiple of the orbital period. The UKF uses the following tuning parameters: α = 1, β = 2, κ = 3 - d = -3, for its sigma points spread. For the AEGIS method, the three-component splitting library is used (AEGIS-3), and the threshold on the allowed deviation of the differential entropy is set as $\Delta H = 0.001 H_0$ [19]. The value of H_0 is unique for each mixture component and based on the covariance at the latter of the last posterior estimate or the output of a splitting operation. After each measurements pass, the AEGIS algorithm is forced to have only one Gaussian component with the posterior mean and

covariance matrix. This simple merging algorithm reduces AEGIS computational burden and adds conservatism that cannot cause divergence (a Gaussian distribution is the most uncertain given any finite covariance matrix). The EnGMF method uses 1000 particles. Both the AEGIS and the EnGMF use the UKF measurement update equations for incorporating measurement information in each GMM component. For the UKF and EnGMF, two implementations with Cartesian coordinates and the equinoctial orbital elements are compared. AEGIS is only implemented in Cartesian coordinates. An AEGIS implementation in equinoctial coordinates will result in very few component splits as the splits occur due to nonlinearity in the propagation, making equinoctial AEGIS very similar to the equinoctial UKF.

These three algorithms are compared based on accuracy, complexity, and consistency. The accuracy of the filters is represented by their root-mean-square error (RMSE), which is computed from the true and estimated states at each measurement update time for all Monte Carlo simulations. The filters' complexity is represented by their average execution time per filtering run in a C++ implementation on a 3.2 GHz single-core Ubuntu operating system. The filters' consistency is examined using the scaled normalized estimation error squared (SNEES) β^R which is defined as follows:

$$\beta_k^R = \frac{1}{Md} \sum_{j=1}^{M} (\boldsymbol{x}_k^{(j)} - \hat{\boldsymbol{x}}_k^{(j)})^{\mathrm{T}} (\boldsymbol{P}_k^{(j)})^{-1} (\boldsymbol{x}_k^{(j)} - \hat{\boldsymbol{x}}_k^{(j)}) \tag{4.23}$$

where M is the number of Monte Carlo simulations, $\boldsymbol{x}_k^{(j)}$ are the true states, $\hat{\boldsymbol{x}}_k^{(j)}$ are the estimated states, $\boldsymbol{P}_k^{(j)}$ are the filter's estimated error covariance matrix of the j-th Monte Carlo run at the time step k. The size of the state space $d = 6$ is used

to scale the NEES value [10] such that a consistent filter will result in a SNEES of one rather than a NEES of d. If the SNEES value is much greater than 1, it means the estimator is divergent; however, if the value is much smaller than 1, it indicates the estimator is too conservative. When the estimator is consistent, SNEES should be nearly one at all times.

The time history of the RMS position errors of the 100 simulations is depicted in Figure 4.2 and the position's RMSE values of each filter are listed in Table 4.2. Due to their nonlinear nature, AEGIS and EnGMF provide better performance than the UKF at the very first measurement update. However, in this measurement-rich environment, equinoctial UKF performs near the top in accuracy, and it is the most consistent at a small fraction of the computational cost of nonlinear filters. From the results, it is also shown that the UKF and EnGMF with equinoctial orbital elements outperforms the corresponding filter with Cartesian coordinates. Nevertheless, the best performance in terms of estimation accuracy is obtained with the AEGIS, closely followed by equinoctial UKF.

Figure 4.3 shows the SNEES value for 100 Monte Carlo simulations and

Table 4.2: Monte Carlo averaged RMSE, SNEES, and computation time for 100 simulations

	Position's RMSE (km)	SNEES	Computation time (sec)
UKF (Cartesian)	0.2212	507.4479	3.36
UKF (Equinoctial)	0.1839	1.2425	3.55
AEGIS-3 (Cartesian)	0.1810	1.5058	460.34
EnGMF (Cartesian)	0.3320	0.4986	189.47
EnGMF (Equinoctial)	0.3284	0.4920	190.49

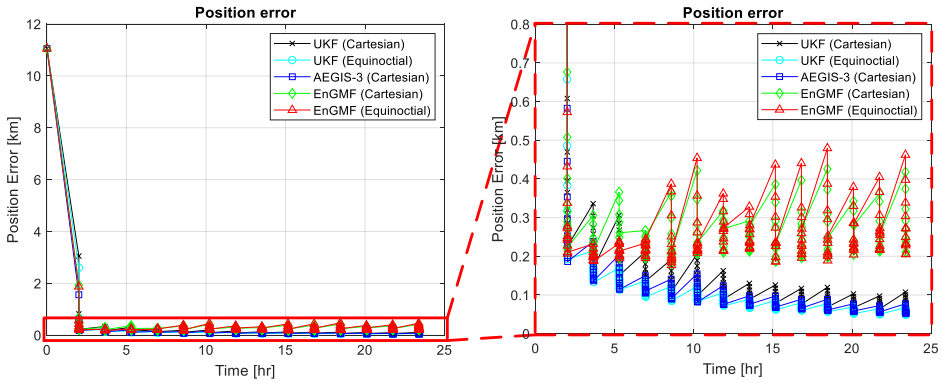
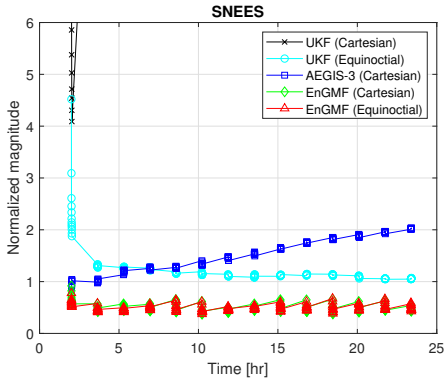


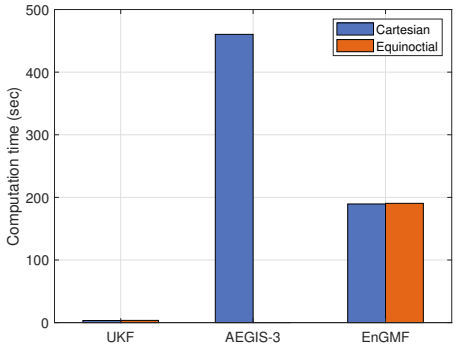
Figure 4.2: The average RMSE for 100 Monte Carlo simulations, 1 pass per orbit

the average computation time per filtering run for all the filters. In Figure 4.3(a), the SNEES value of the EnGMF is smaller than 1, which means the EnGMF is too conservative. For the EnGMF, the covariance matrix calculated by Silverman's rule for each Gaussian component is over-smoothed since the density is not truly Gaussian. The value of the AEGIS filter is gradually increased starting from the value 1. For the UKF, it works better when using equinoctial orbital elements than when using Cartesian coordinates. When the UKF uses Cartesian coordinates, it diverges in two out of 100 simulations, which means the estimate error completely exceeded the ± 3 sigma predicted standard deviations of the posterior covariance matrix. As is typical for linear estimators without underweighting [116], equinoctial UKF is overly optimistic in processing the very first batch of measurements, but due to the measurement-rich scenario, it recovers nicely and achieved very good consistency.

The time-averaged SNEES value to the total samples of 100 cases is listed in Table 4.2. The average computation time is also presented in Table 4.2. In terms



(a) The SNEES value



(b) The average computation time per filtering run

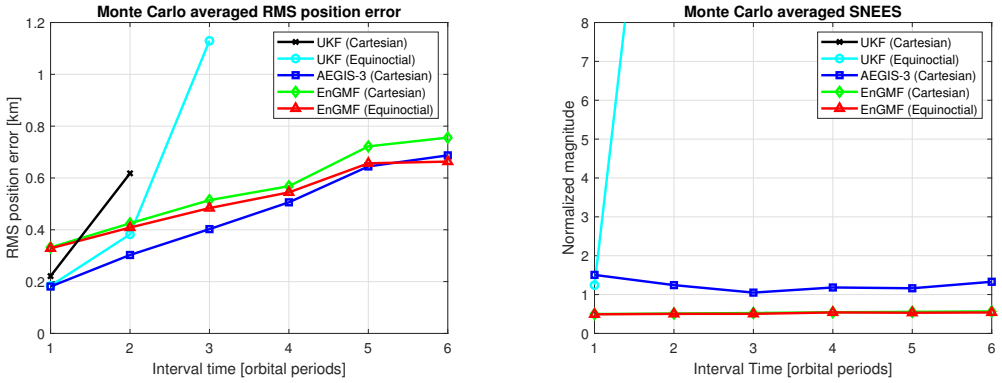
Figure 4.3: The SNEES value and the average computation time per filtering run for 100 Monte Carlo simulations

of computation time, the best performance is obtained with the UKF by a wide margin (as expected from a simple linear filter), and the EnGMF reduces the mean computation time by 58.73% in comparison with the AEGIS. Notice that resetting the GMM in AEGIS to a single component after each measurement pass greatly reduces its computational cost when compared to other merging/pruning schemes.

Having established the baseline performance of the estimators with one measurement pass per orbital period, we focus on the real challenge addressed by this paper: scarcity of measurements. Additional simulations are performed when the gap between measurement passes is increased to 2, 3, 4, 5, and 6 orbital periods. As in the previous case, a Monte Carlo analysis is performed with 100 simulations. As we are concerned with computational speed, we set a maximum allowable number of GMM components for the AEGIS to be 1000 to contain its overall run time, and relaxing this constraint will result in an accurate, but very slow filter.

Figure 4.4(a) displays the position RMSE of the UKF, AEGIS, and EnGMF in all the six cases and the Monte Carlo averaged RMS position errors for all cases are listed in Table 4.3, where n/a indicates the filter has diverged. The time-averaged SNEES value of each estimator for all the six cases is displayed in Figure 4.4(b). The AEGIS outperforms the EnGMF with Cartesian coordinates in terms of RMS accuracy for all the six cases. However, the EnGMF with the equinoctial orbital elements provides better estimation accuracy than the AEGIS when the interval time between measurement passes is 6 orbital periods. Also note that the RMS position error of the AEGIS increases more rapidly with the orbital periods than the EnGMF as shown in Figure 4.4(a). While the equinoctial UKF provides excellent performance for the one-orbit interval period, its performance is severely degraded in terms of accuracy and consistency for the two- and three-orbits case, and is completely diverging for 4–6 orbital periods in-between measurements pass. This is another confirmation that linear dynamics is not sufficient to justify the use of a linear estimator, as nonlinear measurements also need to be addressed.

The choice of using Silverman's rule in the EnGMF rather than performing bandwidth optimization is a trade between speed and accuracy/consistency. However, since the choice results in a conservative filter (estimated covariance larger than actual one) this trade off is deemed worthy when the goal is to maintain custody of a very high number of SOs. The EnGMF implementation in the equinoctial elements provides better and better performance than the EnGMF with Cartesian coordinates as the interval between measurement passes is increasing. The UKF with Cartesian coordinates and the equinoctial orbital elements diverges when the



(a) Monte Carlo averaged RMSE (b) Monte Carlo averaged SNEES value

Figure 4.4: Monte Carlo averaged RMSE and SNEES value for all the six cases

Table 4.3: Monte Carlo averaged RMS position errors for all the six cases

	1	2	3	4	5	6
UKF (Cartesian)	0.2212	0.6178	n/a	n/a	n/a	n/a
UKF (Equinoctial)	0.1839	0.3823	1.1294	n/a	n/a	n/a
AEGIS-3 (Cartesian)	0.1810	0.3027	0.4026	0.5060	0.6445	0.6872
EnGMF (Cartesian)	0.3320	0.4248	0.5144	0.5682	0.7219	0.7559
EnGMF (Equinoctial)	0.3284	0.4086	0.4838	0.5444	0.6566	0.6632

interval time is more than 3 and 4 orbital periods, respectively.

Figure 4.5 shows the computation time of each filter, which is normalized by the value for the EnGMF with Cartesian coordinates. Compared to the AEGIS, the EnGMF reduces the computation time by 59.91% on average.

A more in-depth comparison of the performances of the EnGMF and AEGIS algorithms is shown for the ten-orbits interval periods. Figures 4.6 and 4.7 present the time history of the RMS position errors, SNEES values, and average computation time per filtering run of the EnGMF and AEGIS, and each value is also described in Table 4.4. In terms of accuracy, the EnGMF with Cartesian coordinates

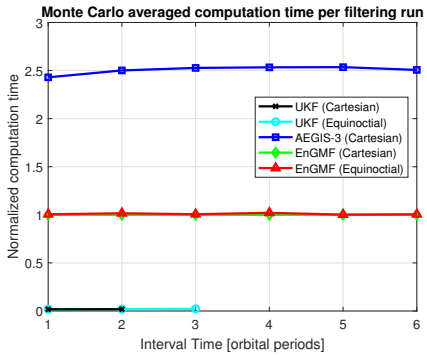


Figure 4.5: Monte Carlo averaged computation time per filtering run

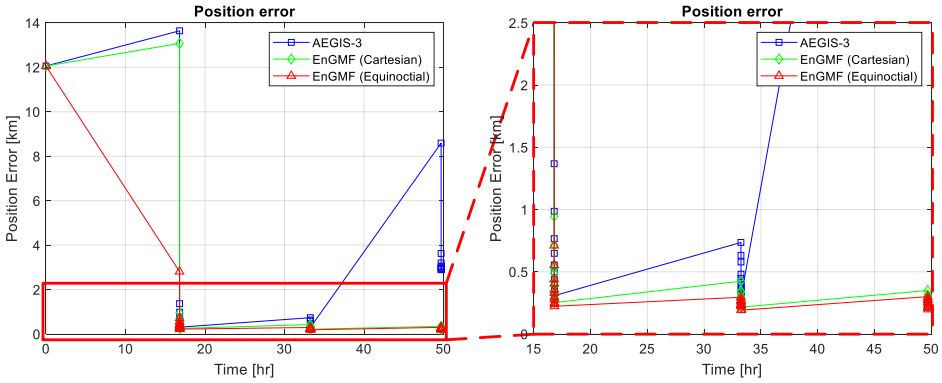
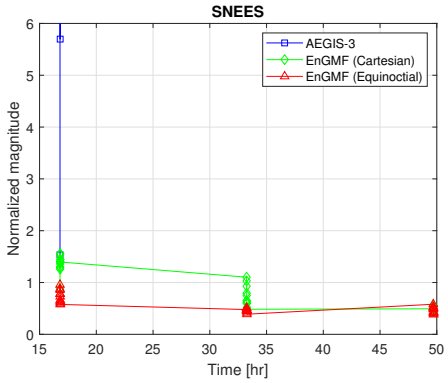


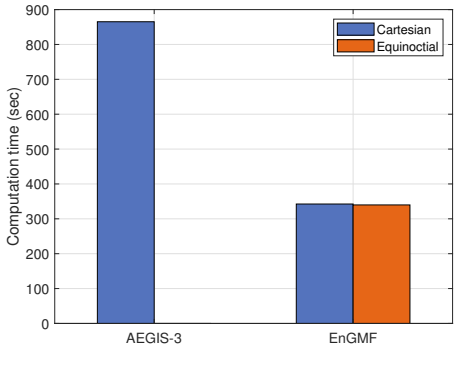
Figure 4.6: The average RMSE for 100 Monte Carlo simulations, the ten-orbits case

or the equinoctial orbital elements outperforms the AEGIS method over time. This is because the AEGIS filter diverges in seven out of 100 simulations as shown in Figure 4.7(a), whereas the EnGMF is conservative. Moreover, Figure 4.7(b) shows that the EnGMF reduces the computation time by 60.57% compared to the AEGIS. In Table 4.4, we can see that the EnGMF using the equinoctial orbital elements obtains the best performance in terms of accuracy and mean computation time.

The performance of the estimators gets worse as the gap between measure-



(a) The SNEES value



(b) The average computation time per filtering run

Figure 4.7: The SNEES value and the average computation time per filtering run for 100 Monte Carlo simulations, the ten-orbits case

Table 4.4: Monte Carlo averaged RMSE, SNEES, and computation time for the ten-orbits case

	Position's RMSE (km)	SNEES	Computation time (sec)
AEGIS-3 (Cartesian)	2.1578	1.6459e+06	865.17
EnGMF (Cartesian)	0.9930	0.8595	342.46
EnGMF (Equinoctial)	0.6688	0.5504	339.86

ment passes increases until it eventually diverges. To evaluate the performance of the EnGMF with the equinoctial elements under a sparser measurement data condition, a Monte Carlo analysis is performed with 100 simulations when the gap between measurement passes is increased to 20 orbital periods. The value of 20 orbital periods is chosen because it causes one divergence out of 100 runs when 1000 particles are used. The analysis is repeated for an EnGMF implementation with 2000 particles. Figures 4.8 and 4.9 show the time history of the RMS position errors, SNEES values, and average computation time of the 100 simulations,

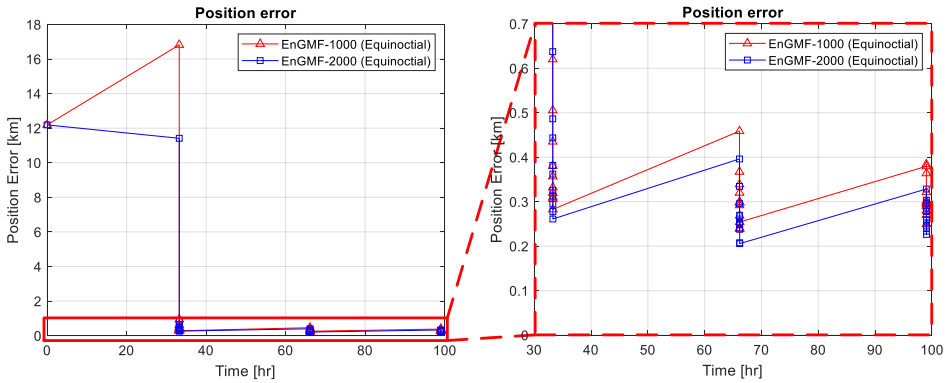
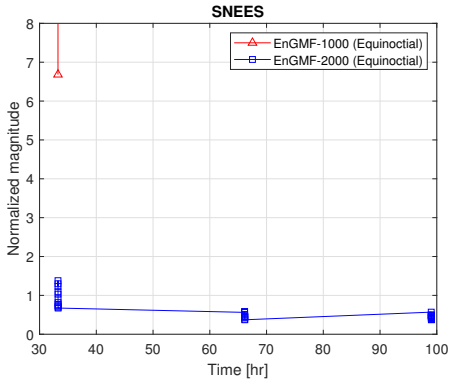
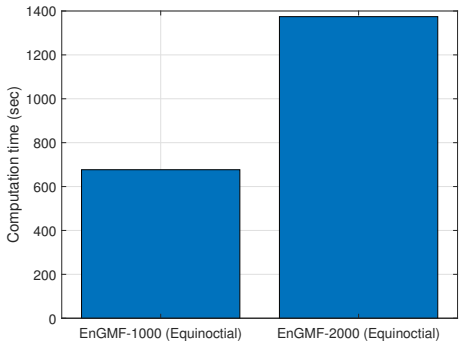


Figure 4.8: The average RMSE for 100 Monte Carlo simulations, the twenty-orbits case

and each value is listed in Table 4.5. The result shows that the EnGMF with 2000 particles outperforms the EnGMF with 1000 particles in terms of the RMSE and SNEES, which is the result of EnGMF with 1000 particles diverging in one out of 100 Monte Carlo simulations as shown in Figure 4.9(a). The EnGMF with 2000 particles, however, requires almost twice the computation time of the EnGMF with 1000 particles. Thus, the choice of the number of particles in the EnGMF is a trade between speed and accuracy/consistency. In other words, even if only a few observation data are available, the EnGMF with a large number of particles can provide accurate and consistent performance tracking SOs in LEO. In future work, we will investigate how to adaptively select an appropriate number of particles for very sparse measurement scenarios.



(a) The SNEES value



(b) The average computation time per filtering run

Figure 4.9: The SNEES value and the average computation time per filtering run for 100 Monte Carlo Simulations, the twenty-orbits case

Table 4.5: Monte Carlo averaged RMSE, SNEES, and computation time for the twenty-orbits case

	Position's RMSE (km)	SNEES	Computation time (sec)
EnGMF-1000 (Equinoctial)	1.1135	989.5970	676.75
EnGMF-2000 (Equinoctial)	0.9372	0.6234	1374.26

4.5 Chapter's Summary

This chapter studies a software-only solution to the orbit determination problem with sparse observation data. The motivation behind the study is the ability to maintain custody of a very large number of low earth orbit objects. As such, it is of outmost importance in this study to strike a balance between estimation accuracy/consistency and computational burden of the methodology employed. A linear filter implementation (unscented Kalman filter) is shown to be inadequate for very scarce measurement scenarios (measurement passes every three orbits or more) re-

gardless of the choice of coordinates (Cartesian or equinoctial orbital elements). A state-of-the-art Gaussian sum filter named AEGIS is shown to perform well at a very high computational cost, but to fail when the number of Gaussian components is artificially capped in order to contain its total execution time.

A newly proposed approach is to modify the kernel-based ensemble Gaussian mixture filter. Each propagated sample of the prior distribution is treated as a Gaussian component with a non-zero covariance matrix. The covariance matrix of a Gaussian component is calculated with Silverman's rule of thumb to reduce the computational cost of numerically optimizing a bandwidth parameter. The rule produces the optimal bandwidth when the samples are drawn from a Gaussian distribution, and results in a conservative estimate for non-Gaussian distributions. Numerical simulations show that the modified algorithm is more accurate and/or faster than the other approaches for sparse measurement scenarios. The conservatism inherent from using Silverman's rule cannot cause filter divergence but can result in slight loss of accuracy. This slight loss of accuracy is deemed an acceptable trade off to computational efficiency for the ultimate purpose of this work: tracking a very large number of space objects. While this conservatism can potentially trigger false collision alarms, an efficient strategy to maintaining a large catalog is using the proposed lower complexity and conservative estimates for the population at-large and to only focus high precision and computationally expensive orbit determination solutions for the very small subset of objects that are deemed at risk for collision.

Chapter 5

Expectation-Maximization Clustering in Particle Gaussian Mixture Filters for Light-Curve Data Processing

5.1 Motivation and Comparison with State of the Art

The algorithms proposed in Chapter 3 and Chapter 4 create a GMM approximation of the distribution by centering a Gaussian component at each and every particle of a PF. At the cost of more computations, the performance of PF/GMM hybrid algorithms can be further improved with a more accurate GMM representation of the prior distribution by choosing parameters of the GMM that optimize some appropriately chosen performance index. This chapter introduces a very accurate filtering algorithm that employs expectation-maximization (EM) clustering to solve an open and very challenging problem in the context of simultaneous space object (SO) tracking and characterization.

Space situational awareness (SSA) refers to knowledge of our near-space environment, including the tracking and identification of SOs orbiting Earth. This task encounters many challenges and one of them is the limited number of sensors available to track and identify an ever growing number of SOs. To extract as much information as possible from the sparse data, sophisticated techniques need to be used to estimate and predict the states of SOs. Precise models of non-gravitational forces acting on SOs are needed for accurate orbit prediction and propagation. Solar Radiation Pressure (SRP) is the main non-gravitational force acting on SOs in or

This chapter is based on: Sehyun Yun and Renato Zanetti, "Nonlinear Filtering of Light-Curve Data," *Advances in Space Research*, 66(7), pp.1672-1688, 2020. The primary content in this work was contributed by the first author.

around Geosynchronous Earth Orbit (GEO) and it can be modeled using the shape and reflectivity properties of the object [50, 68].

Light curve data, which is an object's observed brightness, have been used to analyze attitude observability and to estimate the shape and attitude of SOs [24, 36, 61, 63, 110]. Since light curve observations are sensitive to the object's surface parameters, these can also be estimated from light curve data [57, 109]. Furthermore, it is shown that the space object mass as well as the position, velocity, angle, angular velocity and surface parameters can be estimated by fusing two data types: the angles (line-of-sight) and apparent brightness magnitude of an object [59, 60, 62]. Estimation of these many parameters with relatively little observations, however, has been shown to cause divergence in an UKF when too many states with large uncertainty are estimated simultaneously [108]. Ref. [108] attributes the divergence to information dilution [28].

According to the information dilution theorem (IDT), when additional biases are added to an estimation problem, it is possible that the uncertainties of the original states in the model increase [86]. Moreover, filter divergence may occur because the limited information is not being used in the most proper way [108]. To resolve the information dilution problem in the context of SO tracking, multiple-model adaptive estimation (MMAE) and unscented Schmidt-Kalman filter were implemented to determine which states should be estimated [21, 61, 65, 87]. Ref. [21] quantifies system observability with the information matrix of an estimator and uses the system's observability to determine which states should be estimated. Ref. [65] use an unscented Schmidt-Kalman filter algorithm based on the physical relation-

ship between SRP and observed albedo-area to find low observable states and to consider their contribution to the uncertainty of the system without estimating them. The Schmidt-Kalman filter [91] only estimates a subset of the states, while "considers" the effect of other statess without attempting to infer their value; with this approach the effective number of estimated states is reduced and the effects of information dilution are mitigated. Ref. [87] also employ an unscented Schmidt-Kalman filter algorithm and use the Fisher information matrix (FIM) to measure of the observability of the system; when the FIM becomes close to singular, some states are considered rather than estimated. Considering states in a recursive estimator mitigates information dilution, and to date no study exists that conclusively establishes whether concurrently estimating a dynamic attitude state, angular velocity, and surface parameters from light curve data using a recursive estimator is feasible or if, conversely, information dilution and/or the lack of observability prevent such an estimator to improve knowledge of the system or even avoid divergence. In this chapter, we demonstrate that the principal driver to divergence is the severe nonlinearity of the problem and that it is possible to design a recursive estimator able to improve knowledge of both the attitude and the surface parameters of SOs. The detrimental effects of nonlinearities are exacerbated by information dilution making the UKF design in [108] diverge, but information dilution alone is not cause for divergence; as shown by the algorithms proposed here. In this chapter, the system is studied using three different recursive estimation techniques that successfully estimate all states simultaneously without resorting to only consider the uncertainty of some of them. The three algorithms used are: a newly proposed modification

of PFGMM [84], the truncated interval unscented Kalman filter (TIUKF) [99], and the truncated extended Kalman filter (TEKF) [96].

The first estimation technique used here is based on sequential Monte Carlo methods. Previous works using sequential Monte Carlo methods include [58], where attitude and angular velocity of an SO are estimated from light curve data using a RPF with the generalized Rodrigues parameters used for local attitude error representation. More recently, a marginalized particle filter is used to reduce computational cost of a conventional PF for attitude and angular velocity estimation from light curve data [17, 18]. The use of a PF in this type of problems is particularly appealing for two reasons: i. it provides a nonlinear approximation of the optimal nonlinear estimator, and ii. it handles much larger initial uncertainties than linear estimators (such as the EKF or the UKF); in fact, Ref. [17] assumes an uniform initial attitude uncertainty of almost 360 degrees. Refs. [17] and [18] are very successful in estimating attitude, but do not attempt to concurrently estimate both attitude and surface parameters. It is the addition of surface parameters that causes information dilution and divergence in [108]. This chapter also uses a sequential Monte Carlo filter to estimate both the attitude and rate of the vehicle, but the filter's estimated states include surface parameters and translational states. Surface parameters are successfully estimated in [109] and [57]; the former uses MMAE, essentially choosing between a finite set of possible values for the surface parameters, while the latter is perhaps the closest existing results to this work. The key differences between the two approaches is that Ref. [57] assumes a known (constant) angular velocity, hence the attitude estimation problem can be fully solved

by determining the attitude at a single time; therefore all estimated quantities are constant/static, and a batch approach for Bayesian inverse problems is used by the authors. This work, on the other hand, does not assume *a priori* knowledge of the angular velocity, which is instead estimated together with the attitude and surface parameters in a recursive dynamic filter.

The proposed sequential Monte Carlo method is a modification of the work by [84] but a different clustering algorithm (expectation-maximization, EM) is used to form the GMM density approximation. This modification overcomes issues encountered when applying to this problem two existing particle/GMM hybrid algorithms, [112] and [84]. Ref. [112] introduces a new sequential Monte Carlo algorithm which treats each particle of the pre-propagation distribution as a Gaussian component with a zero or small covariance matrix; the GSF algorithm is used to calculate the posterior distribution. Ref. [84] introduces the PGMF and employs an ensemble of randomly sampled states for the propagation of the conditional state probability density. The propagated ensemble for representing the propagated PDF is clustered using K-means algorithm. While K-means is a simple approach to clustering, it does not produce adequate results for the problem at hand. The K-means algorithm performs a hard assignment of data points to clusters, which means each data point is associated uniquely with one cluster, hence only the points in the same cluster are used to update each mean. Additionally, the K-means algorithm does not account for the covariance. The K-means algorithm can be interpreted as a special case of GMMs clustering in which all mixture weights are equal and the covariance matrices of the mixture components are given by ξI, where ξ is a variance parame-

ter and I is the identity matrix. The EM algorithm for GMMs used in this chapter as it performs a soft assignment based on the posterior probabilities, thus obtaining the proper covariance of the components. In addition to this new nonlinear filter, the modified UKF and EKF are also shown to successfully mitigate the filter divergence issues encountered in the literature while reducing the overall computational complexity of the PFGM with EM Clustering.

The remainder of this chapter is organized as follows. First, the dynamics and measurement models are described and the filter states are presented. Then, the various nonlinear estimation techniques are introduced in Section 5.3. In Section 5.4, simulation results are shown using five filtering algorithms followed by some final remarks of the methodology and results.

5.2 System Models

In this chapter, the inertial position and velocity of SOs are denoted by $\boldsymbol{r}^I = [x\ y\ z]^{\mathrm{T}}$ and $\boldsymbol{v}^I = [v_x\ v_y\ v_z]^{\mathrm{T}}$, respectively. The quaternion, which is based on the Euler axis of rotation $\boldsymbol{n}$ and rotation angle θ, is defined as $\boldsymbol{q} = \left[\sin(\theta/2)\boldsymbol{n}^{\mathrm{T}}\ \cos(\theta/2)\right]^{\mathrm{T}} = \left[\boldsymbol{\varrho}^{\mathrm{T}}\ q\right]^{\mathrm{T}}$ and the angular velocity of the SO with respect to the inertial frame, expressed in body frame, is denoted by $\boldsymbol{\omega}^B_{B/I} = [\omega_x\ \omega_y\ \omega_z]^{\mathrm{T}}$.

5.2.1 Dynamics Model

In this chapter, the orbital dynamics of an SO in ECI coordinates are considered as follows:

$$\ddot{\boldsymbol{r}}^I = -\frac{\mu}{r^3}\boldsymbol{r}^I + \boldsymbol{a}^I_{J_2} + \boldsymbol{a}^I_{srp} \tag{5.1}$$

where μ is the Earth's gravitational parameter, r is the Euclidean norm of $\boldsymbol{r}^I$, $\boldsymbol{a}^I_{J_2}$ is the gravitational perturbation due to non-spherical nature of Earth, and $\boldsymbol{a}^I_{srp}$ is the acceleration perturbation due to SRP.

The J_2 perturbation acceleration equation computes the three component forces in the ECI frame.

$$\ddot{x} = F\left(1 - 5\left(\frac{z}{r}\right)^2\right)\frac{x}{r} \tag{5.2}$$

$$\ddot{y} = F\left(1 - 5\left(\frac{z}{r}\right)^2\right)\frac{y}{r} \tag{5.3}$$

$$\ddot{z} = F\left(3 - 5\left(\frac{z}{r}\right)^2\right)\frac{z}{r} \tag{5.4}$$

where

$$F = -\frac{3}{2}J_2\left(\frac{\mu}{r^2}\right)^2\left(\frac{R_E}{r}\right)^2 \tag{5.5}$$

where J_2 is the second zonal harmonic coefficient and R_E is the Earth's equatorial radius. Higher order spherical harmonics are neglected without loss of generality. At geosynchrounous distances, the J_2 term is small and higher order spherical harmonics are not needed to demonstrate the efficacy of the proposed methodologies.

SRP represents the primary non-gravitational force acting on SOs in GEO and the acceleration due to SRP is modeled using the shape of the body. In this chapter, it is assumed that the shape model consists of a finite number of flat facets; the i^{th} facet is defined by a set of three orthonormal basis vectors $\boldsymbol{u}^B_u$, $\boldsymbol{u}^B_v$, and $\boldsymbol{u}^B_n$ expressed in the body coordinates. The unit vector $\boldsymbol{u}^B_n$ points outward normal of the facet, whereas the vectors $\boldsymbol{u}^B_u$ and $\boldsymbol{u}^B_v$ lie in the plane of the facet. The geometry of the modeled reflection is shown in Figure 5.1. The acceleration perturbation due

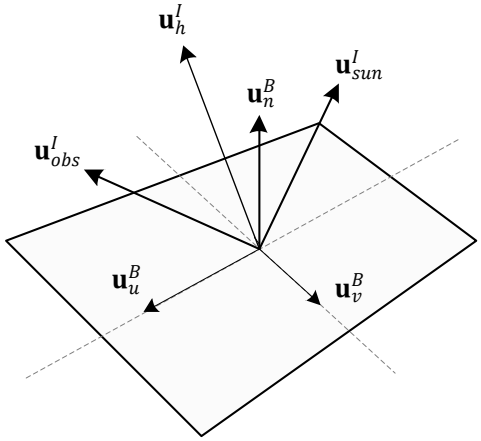


Figure 5.1: Geometry of reflection

to SRP, assuming a Lambertian reflectance model, is then given by [108]:

$$\boldsymbol{a}_{srp}^I = \sum_{i=1}^{N_{facets}} \boldsymbol{a}_{srp}^I(i) \tag{5.6}$$

$$\begin{aligned} \boldsymbol{a}_{srp}^I(i) = &- \frac{S_F \mathcal{A}(i) \left(\boldsymbol{u}_n^I(i) \cdot \boldsymbol{u}_{sun}^I\right)}{mc} \cdot \\ &\left(\left(1 - sF_0\right) \boldsymbol{u}_{sun}^I + \left(\frac{2}{3} d\rho + 2sF_0 \left(\boldsymbol{u}_n^I(i) \cdot \boldsymbol{u}_{sun}^I\right) \right) \boldsymbol{u}_n^I(i) \right) \end{aligned} \tag{5.7}$$

where N_{facets} is the number of facets, S_F is the solar flux, m is the mass of the SO, c is the speed of light, and $\mathcal{A}(i)$ is the area of the i^{th} facet. The unit vector $\boldsymbol{u}_n^I(i)$ is the normal vector pointing outward along the i^{th} surface and $\boldsymbol{u}_{sun}^I$ is the unit vector pointing from the SO to the Sun. Scalars s and d are the fraction of the specular bidirectional reflectance R_s and the diffuse bidirectional reflectance R_d, respectively, where $s + d = 1$. F_0 and ρ are the specular and diffuse reflectance of the facet i at normal incidence, respectively.

As commonly done in aerospace engineering applications [115], the direction cosine matrix is used as the attitude matrix representation in this study. The

relationship between the vector $\boldsymbol{v}^B$ in the body frame and the vector $\boldsymbol{v}^I$ in the inertial frame is described by the attitude matrix $A(\boldsymbol{q})$ such as

$$\boldsymbol{v}^B = A(\boldsymbol{q})\boldsymbol{v}^I \tag{5.8}$$

and the attitude matrix can be parameterized in terms of the quaternion as follows:

$$A(\boldsymbol{q}) = I_{3\times 3} - 2q[\boldsymbol{\varrho}\times] + 2[\boldsymbol{\varrho}\times]^2 \tag{5.9}$$

where

$$[\boldsymbol{a}\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{5.10}$$

is the skew-symmetric matrix representation of the cross product for a vector $\boldsymbol{a}$. The quaternion dynamic equation is given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\Omega \cdot \boldsymbol{q} \tag{5.11}$$

where

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{5.12}$$

is the skew-symmetric form of the body rotation about the inertial frame. The angular velocity dynamic equation is expressed as follows:

$$\dot{\boldsymbol{\omega}}^B_{B/I} = J_{SO}^{-1}\left(\boldsymbol{T}^B_{srp} - \left[\boldsymbol{\omega}^B_{B/I}\times\right] J_{SO}\boldsymbol{\omega}^B_{B/I}\right) \tag{5.13}$$

where J_{SO} is the inertia matrix of the SO and $\boldsymbol{T}^B_{srp}$ is the total torque acting on the SO due to SRP in the body frame. The force due to SRP can be assumed to act on the centroid of each surface. Then, the total torque is calculated by

$$\boldsymbol{T}^B_{srp} = m \sum_{i=1}^{N_{facets}} \left[l^B(i)\times\right] A(\boldsymbol{q})\boldsymbol{a}^I_{srp}(i) \tag{5.14}$$

where $l^B(i)$ is the position vector from the center of the mass of the SO to the geometric center of the i^{th} facet in body frame.

5.2.2 Measurement Model

Angle data in the form of azimuth (az) and elevation (el) are measures used to estimate the states of an SO. The angle observation equations are expressed as follows:

$$az = \tan^{-1}\left(\frac{\rho_E}{\rho_N}\right) \tag{5.15}$$

$$el = \sin^{-1}\left(\frac{\rho_U}{\|\boldsymbol{d}^I\|}\right) \tag{5.16}$$

where

$$\begin{bmatrix}\rho_E\\ \rho_N\\ \rho_U\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\frac{\pi}{2}-\lambda) & \sin(\frac{\pi}{2}-\lambda)\\ 0 & -\sin(\frac{\pi}{2}-\lambda) & \cos(\frac{\pi}{2}-\lambda)\end{bmatrix}\begin{bmatrix}\cos(\frac{\pi}{2}+\theta) & \sin(\frac{\pi}{2}+\theta) & 0\\ -\sin(\frac{\pi}{2}+\theta) & \cos(\frac{\pi}{2}+\theta) & 0\\ 0 & 0 & 1\end{bmatrix}\boldsymbol{d}^I \tag{5.17}$$

where $\boldsymbol{d}^I$ is the position vector from an observer to the SO, $\|\cdot\|$ means the Euclidean norm, θ and λ are the sidereal time and geodetic latitude of the observer, respectively, and $[\rho_E\ \rho_N\ \rho_U]^{\mathrm{T}}$ is the position vector converted from the inertial to the local topocentric East-North-Up coordinates. In this study, light travel time delay is not considered.

Along with the azimuth and elevation, the light curves, which are the time-varying apparent brightness measurements of the SO, are also used. The apparent brightness magnitude measured by the observer is computed by

$$m_{app} = -26.7 - 2.5\log_{10}\left|\sum_{i=1}^{N_{facets}}\frac{f_r(i)\mathcal{A}(i)\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{sun}^I\right)\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{obs}^I\right)}{\|\boldsymbol{d}^I\|^2}\right| \tag{5.18}$$

where -26.7 is the apparent magnitude of the Sun, $\boldsymbol{u}_{obs}^{I}$ is the unit vector pointing from the SO to the observer, and $f_r(i)$ is the bidirectional reflectance distribution function (BRDF) for the i^{th} facet. The BRDF models light distribution of a surface due to incident light and it is a function of two directions, one toward the light source and one toward the observer [9, 61]. The BRDF can be decomposed into a specular component and a diffuse component as follows:

$$f_r(i) = sR_s(i) + dR_d(i) \tag{5.19}$$

The specular reflectance is mirror-like and the diffuse reflectance is Lambertian which means that light is equally reflected in all directions. These bidirectional reflectances are calculated differently for the various models. In this chapter, we use a modified version of the Phong model with a simple form of a non-Lambertian diffuse reflectance [9]. Under the flat facet assumption, the specular bidirectional reflectance is given by

$$\begin{aligned} R_s(i) =& \frac{\sqrt{(n_u(i)+1)\,(n_v(i)+1)}}{8\pi}\cdot \\ & \frac{\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_h^I\right)^{n_u(i)\left(\boldsymbol{u}_u^I(i)\cdot\boldsymbol{u}_h^I\right)^2+n_v(i)\left(\boldsymbol{u}_v^I(i)\cdot\boldsymbol{u}_h^I\right)^2}}{\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{sun}^I+\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{obs}^I-\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{sun}^I\right)\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{obs}^I\right)}F(i) \end{aligned} \tag{5.20}$$

where $n_u(i)$ and $n_v(i)$ are the anisotropic reflectance properties of the i^{th} surface along the $\boldsymbol{u}_u^B(i)$ and $\boldsymbol{u}_v^B(i)$ directions, respectively. Without loss of functionality, in this study, they are assumed to be set equal to each other for the sake of simplicity ($n_u(i) = n_v(i) = n(i)$). Then, Eq. (5.20) is simplified as follows:

$$\begin{aligned} R_s(i) =& \frac{(n(i)+1)}{8\pi}\cdot \\ & \frac{\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_h^I\right)^{n(i)}}{\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{sun}^I+\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{obs}^I-\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{sun}^I\right)\left(\boldsymbol{u}_n^I(i)\cdot\boldsymbol{u}_{obs}^I\right)}F(i) \end{aligned} \tag{5.21}$$

where $\boldsymbol{u}_h^I$ is the normalized half vector which bisects the angle between $\boldsymbol{u}_{sun}^I$ and $\boldsymbol{u}_{obs}^I$:

$$\boldsymbol{u}_h^I = \frac{\boldsymbol{u}_{sun}^I + \boldsymbol{u}_{obs}^I}{\|\boldsymbol{u}_{sun}^I + \boldsymbol{u}_{obs}^I\|} \tag{5.22}$$

and the Fresnel reflectance $F(i)$ is approximated as

$$F(i) = F_0(i) + \left(\frac{1}{s} - F_0(i)\right)\left(\boldsymbol{u}_{sun}^I \cdot \boldsymbol{u}_h^I\right) \tag{5.23}$$

The diffuse bidirectional reflectance is calculated as follows:

$$\begin{aligned} R_d(i) =& \frac{28\rho}{23\pi}(1 - sF_0) \cdot \\ & \left[1 - \left(1 - \frac{\boldsymbol{u}_n^I(i) \cdot \boldsymbol{u}_{sun}^I}{2}\right)^5\right]\left[1 - \left(1 - \frac{\boldsymbol{u}_n^I(i) \cdot \boldsymbol{u}_{obs}^I}{2}\right)^5\right] \end{aligned} \tag{5.24}$$

The apparent magnitude is measured differently mainly depending on the SO attitude and it has the highest value when the surface normal vector $\boldsymbol{u}_n^I$ and the half vector $\boldsymbol{u}_h^I$ are in the same direction. The various values of apparent magnitude depending on the SO attitude are analyzed in [36].

5.2.3 Filter States

In this chapter, it is assumed that the shape of the SO is a cube and each facet of it has the same BRDF surface parameters. The area and mass of the SO are assumed to be known. In addition, the specular reflectance F_0 and diffuse reflectance ρ at normal incidence can be set to be equal to each other because the difference between specular and diffuse reflectance can be expressed by specular s and diffuse fraction parameter d. Thus, the three unique surface parameters to be estimated are n. ρ, d, and they obey the following constraints:

$$n < 0, \quad 0 \leq \rho \leq 1, \quad 0 \leq d \leq 1, \quad s + d = 1$$

Therefore, the state vector utilized is:

$$\boldsymbol{x} = \left[\boldsymbol{q}^{\mathrm{T}} \ \left(\boldsymbol{\omega}_{B/I}^{B}\right)^{\mathrm{T}} \ \left(\boldsymbol{r}^{I}\right)^{\mathrm{T}} \ \left(\boldsymbol{v}^{I}\right)^{\mathrm{T}} \ n \ \rho \ d\right]^{\mathrm{T}} \tag{5.25}$$

5.3 Nonlinear Estimation Techniques for Highly Nonlinear Systems

This section presents the three different estimation algorithms used to analyze the problem at hand: PFGMM, TIUKF, and TEKF.

5.3.1 Particle Filter with Gaussian Mixture Models

The PF with an EM clustering algorithm for GMMs is proposed in this section. A recursive algorithm is used, i.e., knowledge of the distribution $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1})$ at the prior time is assumed and approximated by N i.i.d. samples $\boldsymbol{x}_{k-1}^{(i)}$ such that

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{5.26}$$

where k is an integer that indicates the discrete time step, $\boldsymbol{y}$ is a measurement vector, and $\delta(\cdot)$ is the Dirac delta function. As in the BPF [8], a set of samples at the next time step is generated using the Markov transition kernel $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. Throughout this research, SRP and J_2 are the only perturbations included and additional process noise is neglected. This is a particularly challenging assumption, as particle filters typically rely on process noise to overcome impoverishment.

The next step is to cluster the data into Gaussian mixtures using an EM clustering algorithm. The EM algorithm for GMM approximates the PDF of $\boldsymbol{x}_k$ by

combining several Gaussian components having different means, covariance matrices, and weights. With the i.i.d. data set, the likelihood function for the GMM is expressed by

$$p(\boldsymbol{x}_k|\pi, \boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{N} \sum_{j=1}^{K} \pi_j n(\boldsymbol{x}_k^{(i)}|\boldsymbol{\mu}_j, \Sigma_j) \tag{5.27}$$

where K is a preassigned number of clusters, $n(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$ represents the Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance Σ; and $\boldsymbol{\mu}_j$, Σ_j, and π_j are the means, covariance matrices, and weights of the j^{th} Gaussian component. The PDF's normalization and positivity properties lead to the following constraints on the weights:

$$\pi_j \geq 0, \ \forall j \qquad \sum_{j=1}^{K} \pi_j = 1 \tag{5.28}$$

The goal of the EM clustering algorithm is to maximize the likelihood function with respect to the clustering parameters which are means and covariance matrices of the components, as well as the weights. The algorithm is summarized as follows:

1. Initialize the means $\boldsymbol{\mu}_j$, covariance matrices Σ_j and weights π_j, and evaluate the initial value of the log likelihood.

$$\ln p(\boldsymbol{x}_k|\pi, \mu, \Sigma) = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{K} \pi_j n(\boldsymbol{x}_k^{(i)}|\boldsymbol{\mu}_j, \Sigma_j) \right] \tag{5.29}$$

2. (E step) Evaluate the responsibilities using the current clustering parameter values.

$$\gamma(z_j^{(i)}) = p(z_j^{(i)} = 1|\boldsymbol{x}_k^{(i)}) = \frac{\pi_j n(\boldsymbol{x}_k^{(i)}|\boldsymbol{\mu}_j, \Sigma_j)}{\sum_m^K \pi_m n(\boldsymbol{x}_k^{(i)}|\boldsymbol{\mu}_m, \Sigma_m)} \tag{5.30}$$

where $\gamma(z_j^{(i)})$ is the responsibility of a sample i with respect to a jth Gaussian distribution.

3. (M step) Estimate the new clustering parameters using the current responsibilities to maximize the likelihood. (The following equations are derived in [13])

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} \gamma(z_j^{(i)}) \boldsymbol{x}_k^{(i)}}{\sum_{i=1}^{N} \gamma(z_j^{(i)})} \tag{5.31}$$

$$\boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{N} \gamma(z_j^{(i)}) (\boldsymbol{x}_k^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}_k^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{N} \gamma(z_j^{(i)})} \tag{5.32}$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} \gamma(z_j^{(i)}) \tag{5.33}$$

4. Evaluate the value of the log likelihood and check for convergence of it. If the convergence criterion is not satisfied, replace the old clustering parameters with the new ones and return to step 2.

In this chapter, all components of the GMM are taken with the same covariance matrix. This assumption prevents the GMMs from being too overlapped, while not enforcing hard clustering as in K-means.

Finally, we can incorporate the measurement information by updating the means and covariance matrices of all K components using a Kalman measurement update. The mixture weights need to be updated as well using the components' likelihood functions. We then draw N i.i.d. samples from the posterior distribution; from these samples, we construct a Bayesian estimate and use them as a starting point for the next iteration. The details of the measurement update and the method to draw N i.i.d. samples from a GMM are explained in Chapter 3.

Two approaches to enforce the surface parameters constraints are evaluated.

The first approach is to modify them to unconstrained proxy values. For this study, the same conversion equation used in [108] is applied to convert the surface parameters to the corresponding proxy value and vice versa:

$$p_1 = \ln(n), \quad n = \exp(p_1) \tag{5.34}$$

$$p_2 = \frac{1}{2}\ln\left(\frac{\rho}{1-\rho}\right), \quad \rho = \frac{1}{2}\left(\tanh(p_2) + 1\right) \tag{5.35}$$

$$p_3 = \frac{1}{2}\ln\left(\frac{d}{1-d}\right), \quad d = \frac{1}{2}\left(\tanh(p_3) + 1\right) \tag{5.36}$$

Alternatively, rather than transforming the surface parameters, we can modify the filter to exploit the additional information on the constraints and improve the performance of the filter. In this study, we use the modified rejection-sampling approach which enforces the constraints by simply discarding the particles violating them in the prediction step. Although the number of total samples will be reduced, it is shown that the algorithm maintains the generic properties of the PF [75].

The filter's density, under the assumption of a perfect clustering scheme, converges in probability to the true filter density [84]. The other two approaches studied are based on the constrained UKF and EKF with the PDF truncation approach, which are computationally cheaper and will be presented in the following two subsections.

5.3.2 Truncated Interval Unscented Kalman Filter

The UKF is a linear estimator for nonlinear systems which employs statistical linearization of nonlinear functions through a set of sigma points [54]. The most common schemes to calculate sigma points effectively employs the Gaussian

approximation [45]. The truncated interval unscented Kalman filter (TIUKF) is used in this study to include the inequality constraints on the surface parameters. The TIUKF is composed of two parts: the interval constrained approach which enforces the sigma points interval constraints and the PDF truncation approach which truncates the PDF at the constraint edges [95, 99, 104].

The generic nonlinear dynamics is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{\nu}_k \tag{5.37}$$

where k is the time step, $\boldsymbol{x}_k$ is an $n_{\boldsymbol{x}} \times 1$ vector, $\boldsymbol{f}_k$ is some nonlinear function, and the process noise $\boldsymbol{\nu}_k$ is zero-mean white noise, albeit in this application it will be taken as zero. The measurement is

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k \tag{5.38}$$

where $\boldsymbol{y}_k$ is a measurement vector, $\boldsymbol{h}_k$ is some non-linear function, and $\boldsymbol{\eta}_k$ is the measurement noise consisting of a zero-mean, white sequence with covariance matrix $\boldsymbol{R}_k$, independent from the initial distribution of $\boldsymbol{x}_0$. In addition, assume that the state vector satisfies the interval constraint as follows:

$$\boldsymbol{b}_k \leq \boldsymbol{x}_k \leq \boldsymbol{c}_k \tag{5.39}$$

where $\boldsymbol{b}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{c}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$ are known vectors. If the state vector $\boldsymbol{x}_{i,k}$, where $i = 1, \cdots, n_{\boldsymbol{x}}$, is one-sided, we set $\boldsymbol{b}_{i,k} = -\infty$ or $\boldsymbol{c}_{i,k} = \infty$.

Given an $n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}$ error covariance matrix $P^{xx}_{k|k}$, we generate the $2n_{\boldsymbol{x}} + 1$ sigma points $\mathcal{X}_{k|k}$ holding

$$\boldsymbol{b}_k \leq \mathcal{X}_{j,k|k} \leq \boldsymbol{c}_k, \quad j = 0, \cdots, 2n_{\boldsymbol{x}}. \tag{5.40}$$

To satisfy the inequality constraints, the sigma points are chosen as follows:

$$\mathcal{X}_{k|k} = \hat{\boldsymbol{x}}_{k|k}\mathbb{1}_{1\times(2n_{\boldsymbol{x}}+1)} + [\mathbf{0}_{n_{\boldsymbol{x}}\times 1},\ \theta_{1,k}\text{col}_1\left[S_k\right],\ \cdots,\ \theta_{2n_{\boldsymbol{x}},k}\text{col}_{2n_{\boldsymbol{x}}}\left[S_k\right]] \tag{5.41}$$

where $\hat{\boldsymbol{x}}_{k|k}$ is the *a posteriori* state estimate which is assumed to satisfy the interval constraints at time instant k, $\mathbb{1}_{1\times(2n_{\boldsymbol{x}}+1)}$ is an indicator function,

$$S_k = \left[\left(P^{xx}_{k|k}\right)^{1/2} - \left(P^{xx}_{k|k}\right)^{1/2}\right], \tag{5.42}$$

and

$$\theta_{j,k} = \min\left(\sqrt{n_{\boldsymbol{x}}+\lambda_U},\ \Theta_1,\ \Theta_2\right), \quad \text{for } j = 1,\cdots,2n_{\boldsymbol{x}} \tag{5.43}$$

where, for $i = 1,\cdots,n_{\boldsymbol{x}}$,

$$\Theta_1 = \min_{j:S_{(i,j),k}>0}\left(\infty,\ \frac{c_{i,k}-\hat{\boldsymbol{x}}_{i,k|k}}{S_{(i,j),k}}\right), \quad \Theta_2 = \min_{j:S_{(i,j),k}<0}\left(\infty,\ \frac{b_{i,k}-\hat{\boldsymbol{x}}_{i,k|k}}{S_{(i,j),k}}\right), \tag{5.44}$$

and $\lambda_U = \alpha^2\left(n_{\boldsymbol{x}}+\kappa\right) - n_{\boldsymbol{x}}$ is a scaling parameter [107]. The constant α determines the spread of the sigma points around $\hat{\boldsymbol{x}}_{k|k}$ and it is usually set to a small positive number ($10^{-4} \leq \alpha \leq 1$). κ is a secondary scaling parameter which is usually set to $3 - n_{\boldsymbol{x}}$. Based on the above sigma points, the associated weights are computed as follows:

$$W^m_0 = e_k, \quad W^c_0 = e_k + \left(1-\alpha^2+\beta\right) \tag{5.45}$$

$$W^m_j = W^c_j = d_k\theta_{j,k} + e_k, \quad \text{for } j = 1,\cdots,2n_{\boldsymbol{x}} \tag{5.46}$$

where the constant β is used to include prior knowledge of the distribution of $\boldsymbol{x}$,

and

$$d_k = \frac{2\lambda_U - 1}{2\left(n_{\boldsymbol{x}} + \lambda_U\right)\left(\sum_{j=1}^{n_{\boldsymbol{x}}} \theta_{j,k} - \left(2n_{\boldsymbol{x}} + 1\right)\sqrt{n_{\boldsymbol{x}} + \lambda_U}\right)} \tag{5.47}$$

$$e_k = \frac{1}{2\left(n_{\boldsymbol{x}} + \lambda_U\right)} - \frac{2\lambda_U - 1}{2\sqrt{n_{\boldsymbol{x}} + \lambda_U}\left(\sum_{j=1}^{n_{\boldsymbol{x}}} \theta_{j,k} - \left(2n_{\boldsymbol{x}} + 1\right)\sqrt{n_{\boldsymbol{x}} + \lambda_U}\right)} \tag{5.48}$$

The derivation of the weights equations is described in [104].

Figure 5.2 illustrates how the sigma points of the TIUKF are chosen compared to the sigma points of the conventional UKF in two dimensional system. When the scaling parameters are $\alpha = 1$, $\beta = 2$, and $\kappa = 1$, and the interval constrains are $b_k = [3\ 2]^{\mathrm{T}}$ and $c_k = [8\ 8]^{\mathrm{T}}$, the mean and covariance matrix of the TIUKF are obtained as follows:

$$\hat{\boldsymbol{x}}_{UKF} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \Rightarrow \hat{\boldsymbol{x}}_{TIUKF} = \begin{bmatrix} 4.3717 \\ 4.2587 \end{bmatrix} \tag{5.49}$$

$$\hat{P}^{xx}_{UKF} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \Rightarrow \hat{P}^{xx}_{TIUKF} = \begin{bmatrix} 1.7122 & 0.0962 \\ 0.0962 & 1.8456 \end{bmatrix} \tag{5.50}$$

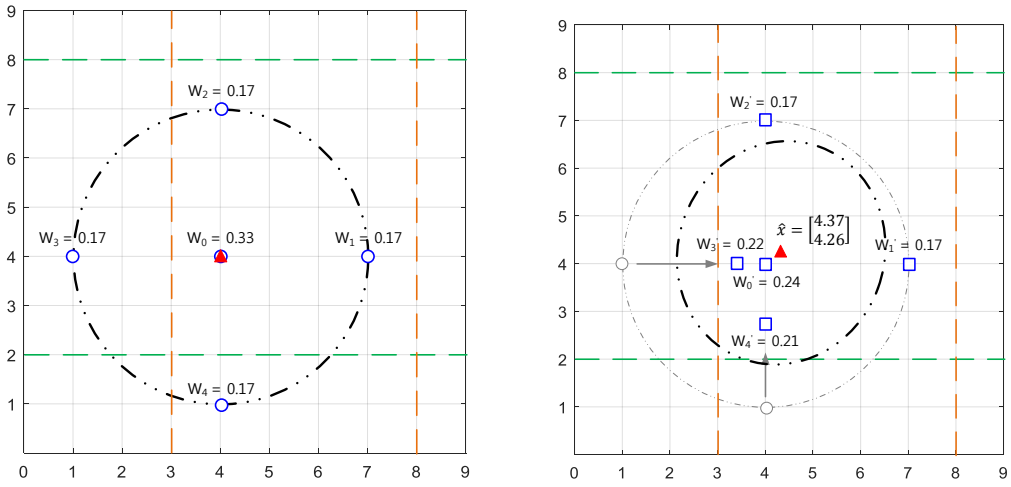
With the above sigma points, the time update equations are the same as the conventional UKF:

$$\mathcal{X}_{j,k+1|k} = \boldsymbol{f}_k(\mathcal{X}_{j,k|k}), \quad j = 0, \cdots, 2n_{\boldsymbol{x}} \tag{5.51}$$

$$\hat{\boldsymbol{x}}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^m \mathcal{X}_{j,k+1|k} \tag{5.52}$$

$$P^{xx}_{k+1|k} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^c \left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right]\left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right]^{\mathrm{T}} \tag{5.53}$$

where $\hat{\boldsymbol{x}}_{k+1|k}$ is the *a priori* state estimate and $P^{xx}_{k+1|k}$ is the *a priori* state estimation error covariance. With the propagated estimates $\hat{\boldsymbol{x}}_{k+1|k}$ and $P^{xx}_{k+1|k}$, a new set of



(a) Sigma points and weights of UKF (b) Sigma points and weights of TIUKF

Figure 5.2: Comparison of the sigma points and weights in UKF and TIUKF

sigma points $\mathcal{X}_{k+1|k}$ of the TIUKF which satisfy the interval constraints and the corresponding weights are recalculated. Then, the measurement update equations are expressed as follows:

$$\mathcal{Y}_{j,k+1|k} = \boldsymbol{h}_{k+1}(\mathcal{X}_{j,k+1|k}), \quad j = 0, \cdots, 2n_{\boldsymbol{x}} \tag{5.54}$$

$$\hat{\boldsymbol{y}}_{k+1} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^m \mathcal{Y}_{j,k+1|k} \tag{5.55}$$

$$P_{k+1|k}^{yy} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^c \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right] \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right]^{\mathrm{T}} + \boldsymbol{R}_k \tag{5.56}$$

$$P_{k+1|k}^{xy} = \sum_{j=0}^{2n_{\boldsymbol{x}}} W_j^c \left[\mathcal{X}_{j,k+1|k} - \hat{\boldsymbol{x}}_{k+1|k}\right] \left[\mathcal{Y}_{j,k+1|k} - \hat{\boldsymbol{y}}_{k+1}\right]^{\mathrm{T}} \tag{5.57}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + P_{k+1|k}^{xy} \left(P_{k+1|k}^{yy}\right)^{-1} \left(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}\right) \tag{5.58}$$

$$P_{k+1|k+1}^{xx} = P_{k+1|k}^{xx} - P_{k+1|k}^{xy} \left(P_{k+1|k}^{yy}\right)^{-1} \left(P_{k+1|k}^{xy}\right)^{\mathrm{T}} \tag{5.59}$$

where $P^{xx}_{k+1|k+1}$ is the *a posteriori* state estimation error covariance, $P^{yy}_{k+1|k}$ is the measurement residual covariance, and $P^{xy}_{k+1|k}$ is the cross covariance.

We then perform the PDF truncation process. The constrained state estimate is the mean of the truncated Gaussian PDF at the constraint edges. The state estimate is normalized in a way that its components are statistically independent of each other to reduce computational effort to determine the truncated PDF. The part of the Gaussian PDF which is outside of the constraints then is removed. After all the constraints are sequentially applied to the corresponding component, we then revert the normalization process to obtain the constrained state estimate. The details of the algorithm are explained in [95] and [96].

5.3.3 Truncated Extended Kalman Filter

The EKF is a nonlinear approximation of the Kalman filter that can be applied to nonlinear systems using the same Kalman filtering framework. Given the system model, Eq. (5.37) and Eq. (5.38), the time update equations are described as follows:

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_{k|k}) \tag{5.60}$$

$$P^{xx}_{k+1|k} = F_k P^{xx}_{k|k} F_k^{\mathrm{T}} \tag{5.61}$$

$$F_k = \left.\frac{\partial \boldsymbol{f}_k(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k}} \tag{5.62}$$

where F_k is the Jacobian of the dynamics evaluated at the posterior mean $\hat{\boldsymbol{x}}_{k|k}$. The measurement update equations are:

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + K_{k+1}\left(\boldsymbol{y}_{k+1} - \boldsymbol{h}_{k+1}(\hat{\boldsymbol{x}}_{k+1|k})\right) \tag{5.63}$$

$$P^{xx}_{k+1|k+1} = P^{xx}_{k+1|k} - K_{k+1}W_{k+1}K^{\mathrm{T}}_{k+1} \tag{5.64}$$

$$K_{k+1} = P^{xx}_{k+1|k}H^{\mathrm{T}}_{k+1}W^{-1}_{k+1} \tag{5.65}$$

$$W_{k+1} = H_{k+1}P^{xx}_{k+1|k}H^{\mathrm{T}}_{k+1} + R_{k+1} \tag{5.66}$$

$$H_{k+1} = \left.\frac{\partial \boldsymbol{h}_{k+1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}} \tag{5.67}$$

where H_{k+1} is the Jacobian of the measurement evaluated at the prior mean $\hat{\boldsymbol{x}}_{k+1|k}$, K_{k+1} is the Kalman gain, and W_{k+1} is the measurement residual covariance. The PDF truncation step which is explained in the previous subsection is then applied to the truncated extended Kalman filter (TEKF) [96].

Despite of the additional information on the constraints, the severe nonlinearities of the system can lead to divergence of the TEKF. For example, the approximation error caused by truncating the nonlinear functions to the first-order (e.g. Eq. (5.62) and Eq. (5.67)) can be significant. It is well-known that when measurement noise is small while the *a priori* uncertainty of the state estimate is relatively large, nonlinear effects can become very significant [67, 116].

To analyze nonlinear effects on the measurement update in detail, a Gaussian second-order filter is considered which includes the second-order terms in the Taylor series expansion [67]. The Kalman gain and measurement residual covari-

ance in the Gaussian second-order filter are expressed as follows:

$$K_{k+1}^{2nd} = P_{k+1|k}^{xx} H_{k+1}^{\mathrm{T}} \left(W_{k+1}^{2nd}\right)^{-1} \tag{5.68}$$

$$W_{k+1}^{2nd} = H_{k+1} P_{k+1|k}^{xx} H_{k+1}^{\mathrm{T}} + R_{k+1} + B_{k+1} \tag{5.69}$$

where matrix B_{k+1} is the contribution of the second-order effects and the ij^{th} component of B_{k+1}, under the Gaussian approximation, is given by

$$B_{ij,k+1} = \frac{1}{2} \operatorname{trace} \left(\left. \frac{\partial^2 \boldsymbol{h}_{i,k+1}(\boldsymbol{x})}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathrm{T}}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}} P_{k+1|k}^{xx} \left. \frac{\partial^2 \boldsymbol{h}_{j,k+1}(\boldsymbol{x})}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathrm{T}}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}} P_{k+1|k}^{xx} \right) \tag{5.70}$$

where $\boldsymbol{h}_{i,k+1}$ is the i-th component of $\boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1})$. Comparing the measurement residual covariance for the EKF in Eq. (5.66) with the measurement residual covariance for the Gaussian second-order filter in Eq. (5.69) and observed the Gaussian second-order filter gain is smaller than the standard EKF gain when the contribution of the second-order term is significant. Consequently, the state estimation error covariance (Eq. (5.64)) of the standard EKF decreases more quickly than the actual state error covariance when the contribution of the second-order term is not negligible.

The Gaussian second-order filter is rarely used in practice due to its reliance on the Gaussian approximation. An alternative method to compensate for the high-order effects that allows for tuning are implemented in this chapter: Lear's underweighting method [116]. Lear's approach to underweighting the measurement is to add a percentage of the *a priori* estimation error covariance to the measurement residual covariance:

$$W_{k+1}^{U.W} = H_{k+1} P_{k+1|k}^{xx} H_{k+1}^{\mathrm{T}} + R_{k+1} + \beta_{UW} H_{k+1} P_{k+1|k}^{xx} H_{k+1}^{\mathrm{T}} \tag{5.71}$$

where β_{UW} is a tuning parameter. The additional term, $\beta_{UW} H_{k+1} P^{xx}_{k+1|k} H^{\mathrm{T}}_{k+1}$, in the measurement residual covariance decreases the Kalman gain, thus reducing the state estimate and *a posteriori* state estimation error covariance update.

Another approach to make the filter more robust in the presence of high uncertainty and nonlinearities is the consider Kalman filter [111]. The effects of highly nonlinear states of the system can be "considered" only, meaning the states are not updated in the filter. In other words, we only update the state estimates which are not highly nonlinear and the corresponding error covariance based on the uncertainty of the highly nonlinear states. The consider Kalman filter algorithm and derivation are explained in [111]. In this chapter, both methods are applied to the TEKF only when the contribution of the *a priori* estimated state uncertainty to the residual covariance is much larger than the measurement noise covariance, i.e. $H_{k+1} P^{xx}_{k+1|k} H^{\mathrm{T}}_{k+1} \gg R_{k+1}$, which is a strong indicator that nonlinear effects might become important [116].

5.4 Numerical Results

For the state estimation problem described in Section 5.2, we adopted the scenario used in [108] to investigate the divergence and accuracy achievable by recursive estimators, i.e., non-batch. Ref. [108] suggests divergence is due to information dilution, as the available data are scarce and not used in the most appropriate way. We concur with this analysis in that information dilution coupled with severe nonlinearities causes divergence in the UKF. However, information dilution alone is not responsible for divergence as a linear system does not exhibit divergence. The

numerical results in this section show that it is possible to design recursive estimators for this problem that do not diverge and that improves the accuracy of all states. It is not only possible with sophisticated sequential Monte Carlo methods, but also with an UKF or even an EKF when appropriate precautions are taken.

In the simulation, a SO is in a geosynchronous orbit with the following orbital elements: $a = 42,364.16932$ km, $e = 0$, $i = 30^\circ$, $M_0 = 91^\circ$, and $\omega = \Omega = 0$. The simulation epoch is 15-March-2010 at 04:00:00 UT and the SO does not pass through the shadow of the Earth during the simulation time. The shape of the SO is a cube with side length 1m and a mass of 2kg and it is assumed that there is no self shadowing in the model. Apparent brightness magnitude and angle measurements are simulated using a ground station located at the top of Haleakala in Maui (latitude $= 20.71^\circ$, longitude $= -156.26^\circ$, and altitude $= 3.5086$km). Measurements are corrupted by additive zero-mean Gaussian white noise with standard deviations of 0.1 for the brightness magnitude and 10 arc-seconds on the azimuth and elevation observation. Both measurements are available every 2 seconds for two hours. The changes we made in this simulation scenario with respect to [108] are that (1) we used the azimuth and elevation observations for angle data instead of right ascension and declination observation and (2) we omitted thermal radiation pressure (TRP) in the dynamics as it did not change the results appreciably. The details of the initial truth state, the initial estimated state, and the initial uncertainty are listed in Table 5.1. Note that the goal of this study is to estimate attitude, attitude rate, and surface parameters simultaneously, and we do so using the initial conditions highlighted in [108] that include an initial attitude uncertainty of 10 degrees

Table 5.1: Initial conditions

State	Initial Truth	Initial Estimate	Uncertainty (1σ)
$\boldsymbol{q}$	0.754 0.133 0.000 0.643	0.695 0.134 0.010 0.706	3.33 deg
$\boldsymbol{\omega}_{B/I}^{B}$ (rad/s)	0.00200 -0.00100 0.00500	0.00212 -0.00106 0.00506	1.16×10^{-4}
$\boldsymbol{r}^{I}$ (km)	-739.4 36682.9 21178.9	-789.4 36732.9 21278.9	100
$\boldsymbol{v}^{I}$ (km/s)	-3.0669 -0.0464 -0.0268	-3.0169 0.0536 -0.0768	0.10
n	150	120	30
ρ	0.40	0.10	0.30
d	0.70	1.00	0.30

(3σ) and that cause their UKF design to diverge. This is a challenging scenario as all surfaces of the SO are assumed to have the same parameters, hence different surfaces are indistinguishable from one another and very large initial attitude errors cannot be resolved. In addition, the assumption of having the same parameters makes the total torque acting on the SO due to SRP become zero. Other studies, including [17, 58] assume known surface parameters with different values for each facet. Under those conditions, a particle filter is able to resolve initial orientation uncertainties much larger than 10 degrees.

The first goal is to investigate whether information dilution alone can cause divergence, or if a nonlinear filter can be successfully applied to this problem. Many nonlinear algorithms such as various flavors of the particle filter as well as PGMF

from [84] and the sequential Monte Carlo filter from [112] were used and they all diverged. The UKF also diverges. These failures are due to the high nonlinearities of the light-curve data combined with the absence of process noise. Divergence is not an intrinsic property of the system, as the modified PGMF algorithm proposed here, named PFGMM1, is able to prevent the divergence of the state. Figure 5.3 shows the position, attitude, and surface parameter errors with the corresponding 3σ predictions when PFGMM1 is used. Notice that proxy values are used in order to estimate the surface parameter without any constraint on their values. All errors are consistent with the uncertainties, meaning that the filter does not diverge. From the analysis, it is shown that it is the severe nonlinearities coupled with the weak observability of the system that leads to divergence, not information dilution. The PFGMM uses 3 clusters with 10,000 particles and for this and all subsequent filters the modified Rodrigues parameters (MRPs) are used to define the local error for the attitude estimation.

The simulation is conducted with the five nonlinear filters described in the previous section: (1) the particle filter with Gaussian mixture models (PFGMM) without the constraint information (PFGMM1), (2) the PFGMM with the constraint information (PFGMM2), (3) the TIUKF, (4) the TEKF with underweighting (TEKF1), and (5) the TEKF with considering parameters (TEKF2).

The introduction of proxy surface parameters, while making the state space unconstrained, adds more nonlinearities to the systems. Alternatively, the constraint can be used as additional information in the modified rejection-sampling algorithm, we denote this filter as PFGMM2. The time history of the state errors and respective

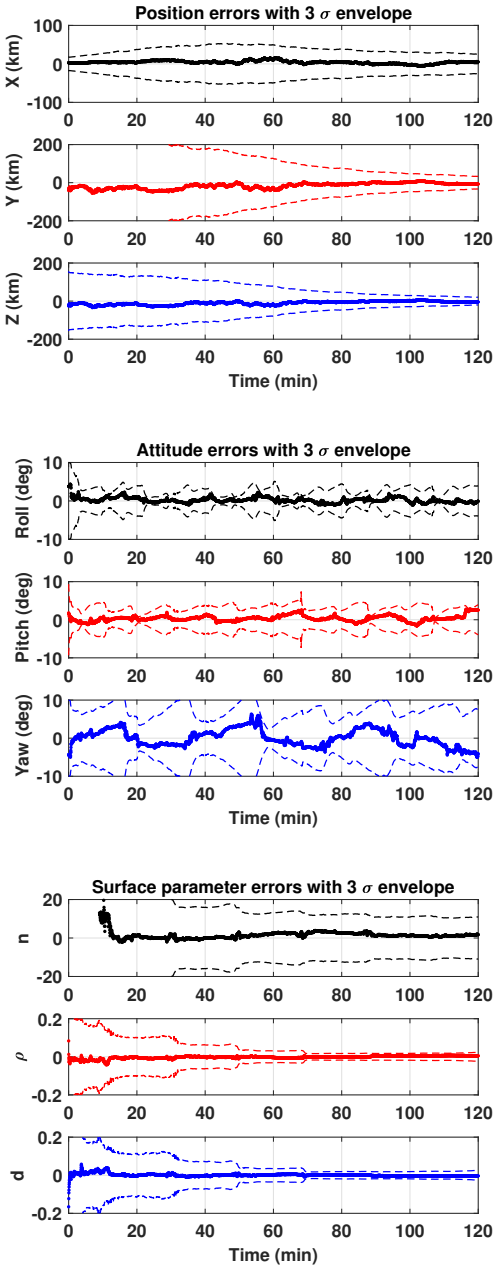


Figure 5.3: Position, attitude, and surface parameter errors with the PFGMM without the constraint information

3σ predicted performance when using the PFGMM2 with the modified rejection-sampling approach is depicted in Figure 5.4. Since the constraint information is added to the estimator, it can be seen that the PFGMM2 has the better performance than the PFGMM1.

The two nonlinear filters proposed (PFGMM1 and PFGMM2) establish that information dilution due to few measurements and many estimated quantities does not necessarily cause filter divergence, and that treating constraints as source of information improves the performance of the filter. The next objective of this investigation is to design a consistent linear estimator, i.e., Kalman filter, which, while producing less accurate estimates than the nonlinear filters above, still produces a consistent, non-diverging solution. The algorithms used are the modifications of the UKF and EKF described in the previous sections.

The TIUKF uses the following tuning parameters: $\alpha = 0.8$, $\beta = 3$, and $\kappa = 3$, for its sigma points spread. The underweighting tuning parameter for the TEKF with underweighting approach (denoted as TEKF1) is $\beta_{UW} = 2.0$. The third and last linear estimator considered is the TEKF with considering parameters (denoted as TEKF2) which treats the surface parameters (since they are highly nonlinear states in the system) as considered states when high nonlinearities are detected. High nonlinearities are declared when $H_{k+1}P^{xx}_{k+1|k}H^{\mathrm{T}}_{k+1} > 3R_{k+1}$ in the brightness magnitude measurement only, as it is the nonlinear measurement that causes divergence.

The simulation results of the three linear filters are shown in Figure 5.5 to Figure 5.7. Comparing the error and covariance of all the cases, the performance of

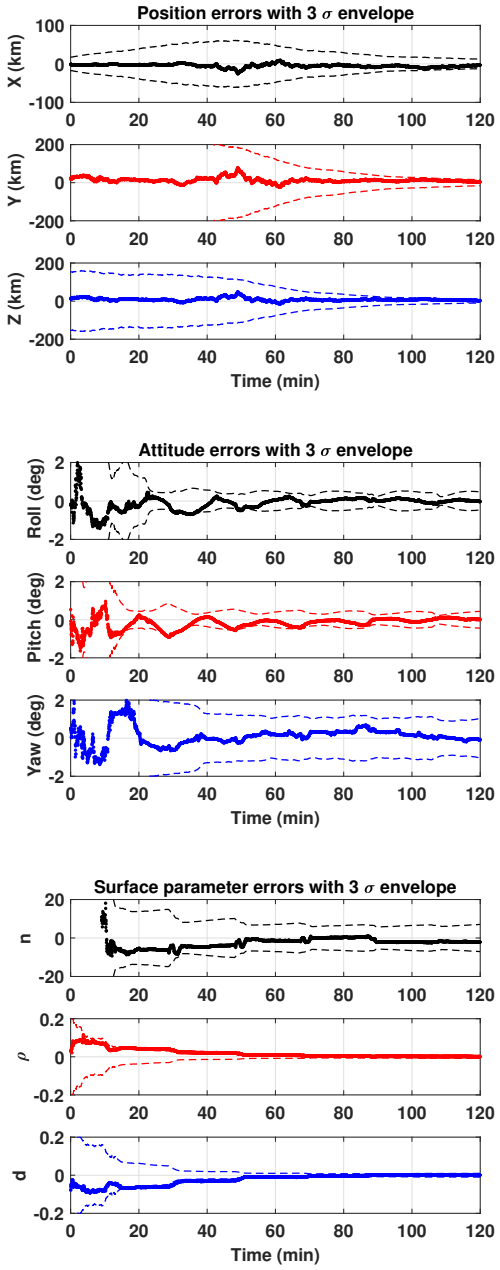


Figure 5.4: Position, attitude, and surface parameter errors with the PFGMM with the constraint information

those filters is comparable to that of the PFGMMs, yet at a reduced computational cost. Based on the criterion, $H_{k+1}P^{xx}_{k+1|k}H^{\mathrm{T}}_{k+1} > 3R_{k+1}$, the TEKF1 used the underweighting parameter as follows: $\beta_{UW} = 2.0$ for $t \leq 16$ and $\beta_{UW} = 0$ for $t > 16$, where t is the simulation time. With the same criterion, the TEKF2 considers surface parameters when $t \leq 48$ and estimates all the states for the rest of the time. Since the PDF truncation step was performed in the TEKF with the methods to compensate for the high-order effects, the uncertainties of the surface parameters and associated states (i.e. attitude) eventually converge to slightly smaller values than those of the PFGMM1.

Table 5.2 lists the time-averaged root mean square error (RMSE) for a single simulation. The best performance is obtained with the PFGMM2 when comparing the time-averaged RMSE. In terms of the RMS attitude, angular velocity, and parameter errors, the TEKFs have the better performance than the PFGMM1, which indicates it is possible to improve the PFGMM by increasing the number of particles and clusters. While an increases in the number of particles and clusters might improve the PFGMM, such an increase would increase the computational cost substantially. The computation time for filtering run in MATLAB on a 3.2 GHz hexa-core Windows operation system is also presented in Table 5.2. In terms of computation time, the TEKFs are the best performers while retaining roughly the same accuracy as the PFGMMs.

As results from single runs cannot definitively assess the performance of a stochastic estimator, a Monte Carlo analysis is performed with 100 simulations to compare the performance of the five filters based on the RMSE and the NCI. The

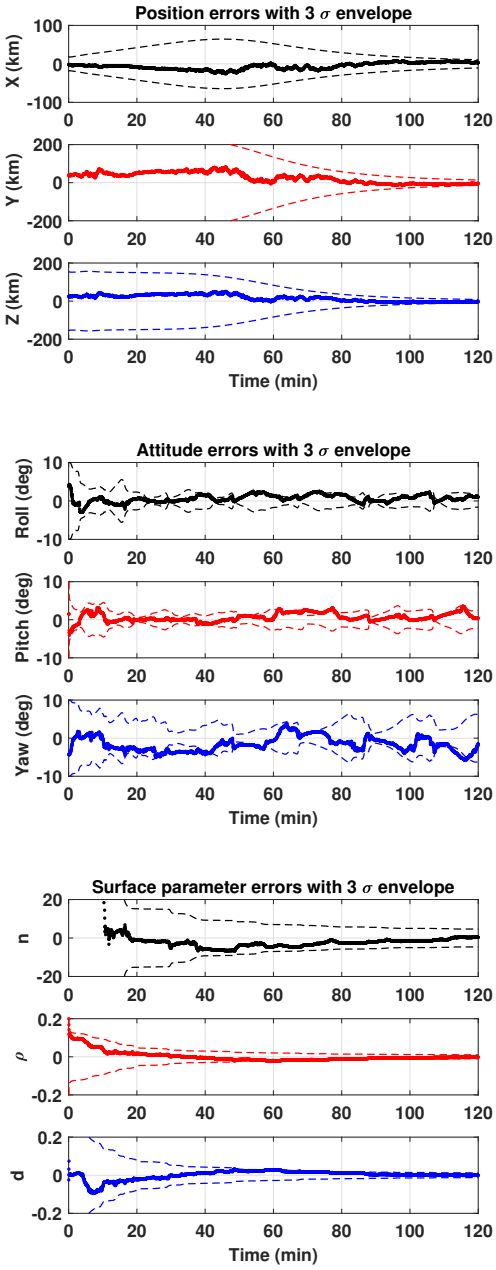


Figure 5.5: Position, attitude, and surface parameter errors with the TIUKF

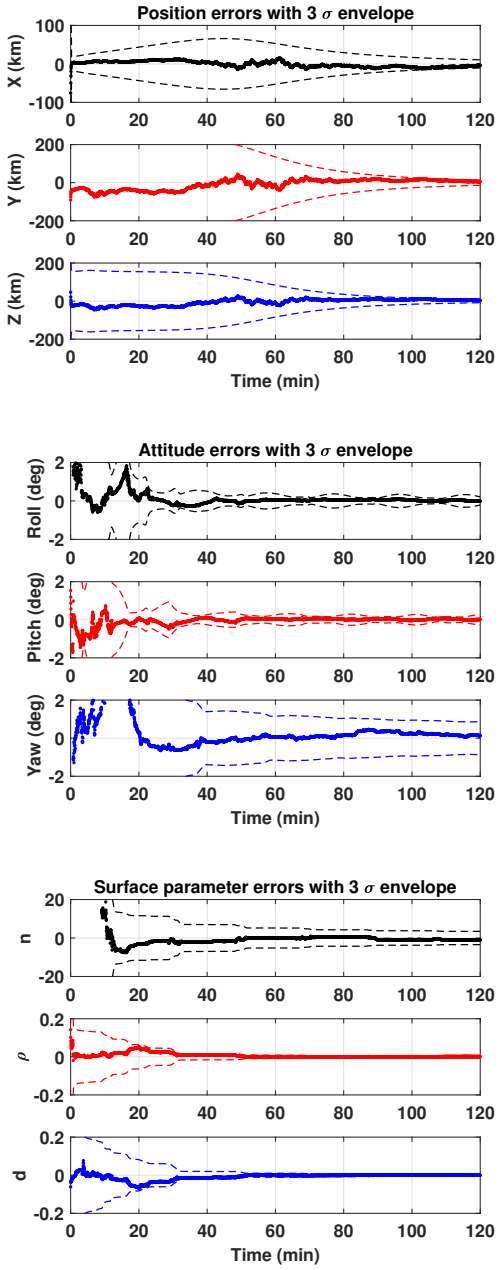


Figure 5.6: Position, attitude, and surface parameter errors with the TEKF with underweighting

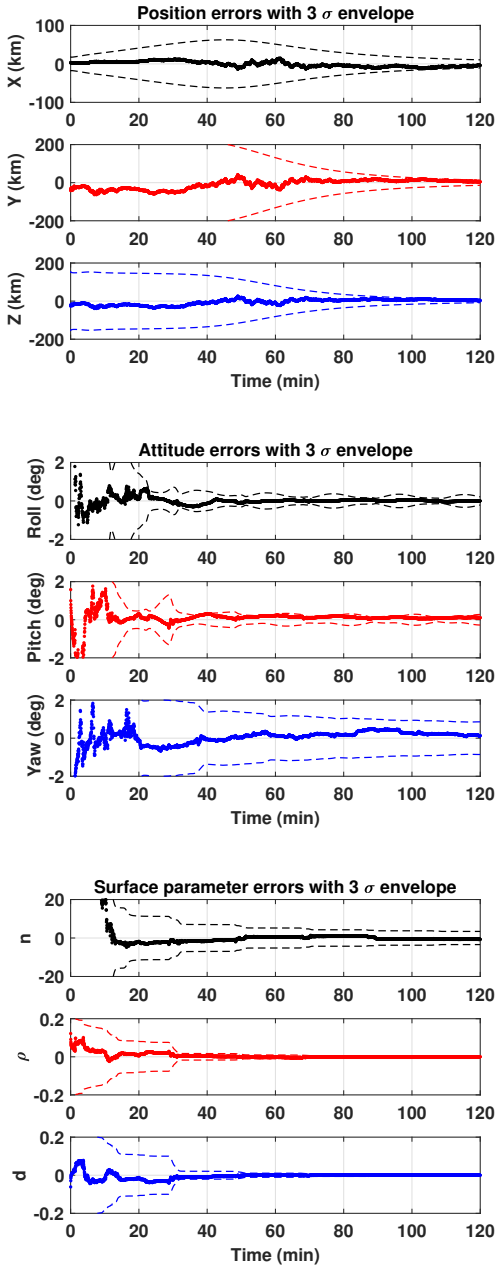


Figure 5.7: Position, attitude, and surface parameter errors with the TEKF with considering surface parameters

Table 5.2: RMSE for a single simulation

Filter	Time-averaged RMSE					Computation time (s)
	Position (m)	Velocity (m/s)	Attitude (deg)	Angular velocity (deg/hr)	Parameter	
PFGMM1	26383.9	22.313	2.440	19.916	8.721	4090.90
PFGMM2	22208.3	15.634	0.706	7.641	8.033	4292.72
TIUKF	45548.6	21.179	3.022	46.934	10.108	158.06
TEKF1	33062.5	22.915	1.121	9.154	8.421	127.87
TEKF2	29865.8	22.336	0.794	8.983	8.557	127.84

RMSE is calculated from the true and estimated states at each time k for each Monte Carlo simulation and the NCI metric is used to measure the consistency of the five estimators. When the difference between the ensemble error covariance matrix of the estimates and the filter's error covariance matrix is small, the NCI value should be zero or nearly zero at all times.

Figure 5.8 to Figure 5.10 display the time history of the RMS position, velocity, attitude, angular velocity, and parameter errors of the 100 simulations. The time-averaged value of the RMSE of all the filters are listed in Table 5.3. The surface parameters of the highly nonlinear light curve measurements are associated with the attitude and angular velocity. It is shown in the figures that the results of the RMS attitude, angular velocity, and surface parameter (ρ and d) errors are highly correlated to each other. The best performance is obtained with the PFGMM2. The results also show that although the performance of the TEKF1 and PFGMM2 are comparable, the TEKF2 has the worst performance among the five filters. In terms of the RMS position, velocity, and surface parameter (n) errors, the performances

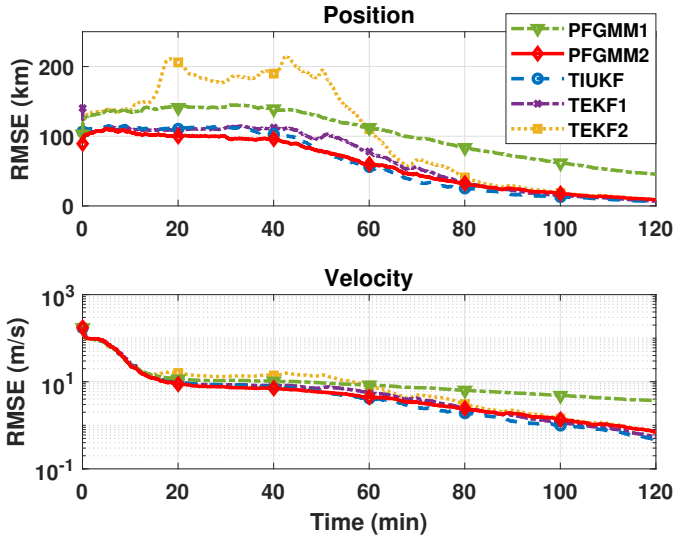


Figure 5.8: Monte Carlo averaged RMSE of position and velocity for 100 random realizations

of the TIUKF and PFGMM2 are comparable. However, the RMS attitude, angular velocity, and surface parameter (ρ and d) errors of the TIUKF are higher than those of any filters, which means the TIUKF is adversely affected by the severe nonlinearities of the system. On the other hand, the RMS attitude, angular velocity, and surface parameter errors of the PFGMM1 which does not use the constraint information are comparable to the PFGMM2 while the RMS position and velocity errors of the PFGMM1 are higher than those of any filters.

The consistency test result of each filter represented by the absolute NCI value is shown in Figure 5.11. In this figure, the NCI values of the PFGMM1 and PFGMM2 are smaller than those of other linear filters and they approach to zero as time passes. The absolute NCI value of the TEKF1 is smaller than that of the TEKF2, which means that Lear's underweighting method is more effective to com-

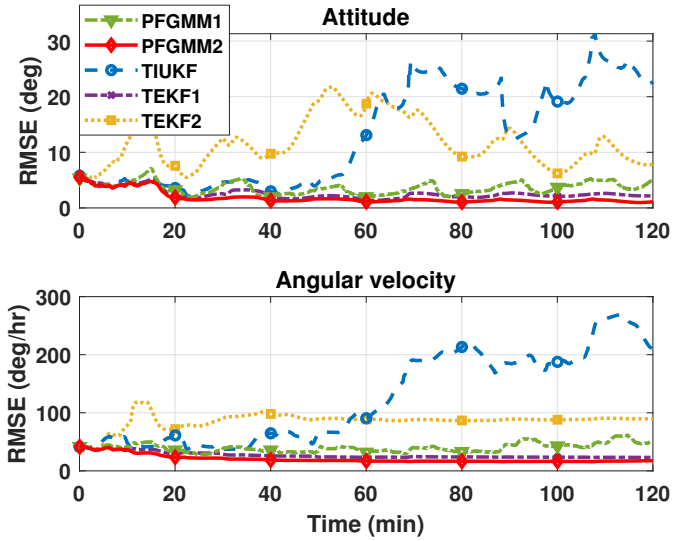


Figure 5.9: Monte Carlo averaged RMSE of attitude and angular velocity for 100 random realizations

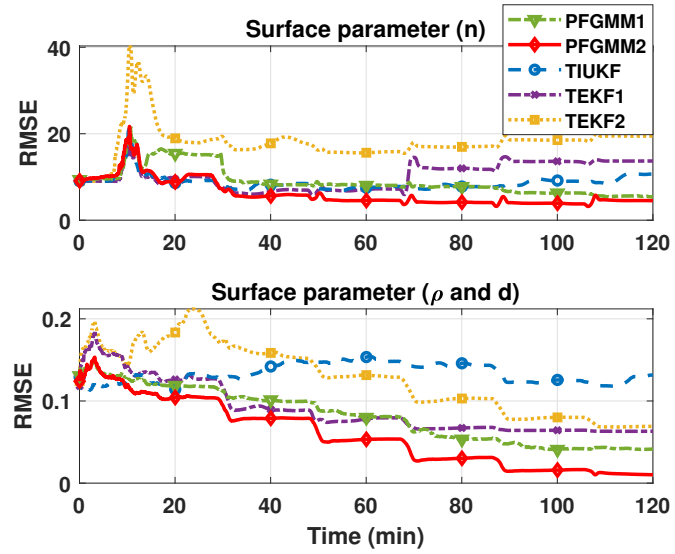


Figure 5.10: Monte Carlo averaged RMSE of surface parameters for 100 random realizations

Table 5.3: RMSE for 100 Monte Carlo simulations

Filter	Time-averaged RMSE					NCI	Computation time (s/simulation)
	Position (m)	Velocity (m/s)	Attitude (deg)	Angular velocity (deg/hr)	Parameter		
PFGMM1	10473.0	13.38	3.57	39.82	8.97	6.10	3886.94
PFGMM2	6058.0	10.52	1.82	20.39	6.16	4.61	3724.63
TIUKF	6128.7	10.29	12.92	124.88	8.69	14.59	120.77
TEKF1	6768.3	11.22	2.55	26.92	10.32	11.90	114.70
TEKF2	10330.0	13.09	11.98	86.20	18.02	15.54	112.64

pensate the nonlinear effects of this system than considering the surface parameters. Moreover, the figure shows that the TIUKF is unsuitable for such a highly nonlinear system. The average computation time per filtering run as well as the time-averaged absolute NCI value are listed in Table 5.3.

5.5 Chapter's Summary

This chapter presents a detailed study of the estimation of the translational and rotational states of near-geosynchronous objects from bearing angles and light curve data. Three parameters of the highly nonlinear light curve measurements are also estimated. The high nonlinearity and weak observability of the system makes this problem particularly challenging for recursing filtering algorithms. This fact is exacerbated by the absence of process noise, which is typically needed to overcome particle impoverishment in particle filters.

A novel approach to nonlinear estimation combining particle filters and Gaussian sum filters using an expectation-maximization clustering method is pro-

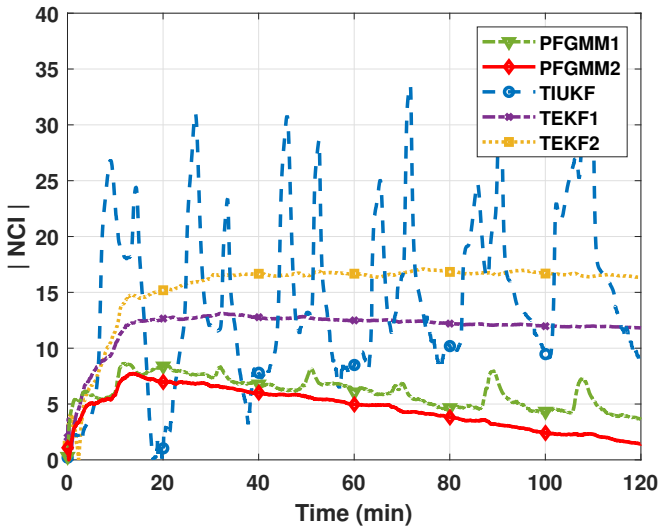


Figure 5.11: Time history of the absolute NCI value for 100 random realization

posed. The advantage of this algorithm is that the use of soft clustering gives a more accurate Gaussian mixture model representation of the prior probability density function over existing related approaches. The soft-clustering approach allows the filter to converge, while a similar existing algorithm using K-means clustering diverges under the conditions of the example studied. The soft clustering works in a way that each point is assigned to all the clusters with different weights or probabilities, thus obtaining the proper covariance of the components. By designing a consitent filter with the same members of the state space and same measurements, it is shown that dilution of information is not a cause of divergence per-se, rather divergence of prior approaches are due to the severe nonlinearities of the system coupled with large initial uncertainties and the weak observability introduced by information dilution.

Finally, three linear estimators were designed and shown to provide good performance. The truncated interval unscented Kalman filter uses the constraint information in the time and measurement update steps and truncates the probability density function after the measurement update. The truncated extended Kalman filter includes not only the probability density function truncation approach but also two extra methods to compensate nonlinear effects.

Chapter 6

Clustering Methods for Particle Filters with Gaussian Mixture Models

6.1 Motivation and Comparison with State of the Art

The particle filter with Gaussian mixture models (PFGMM) proposed in the previous chapter is able to successfully estimates the states of a highly nonlinear system in a challenging application where other filters such as the GSF, BPF, APF, and RPF diverge. In the PFGMM, the propagated samples are clustered to recover a GMM representation of the prior PDF. Ref. [84] demonstrated that the filter's density converges in probability to the true filter density under the assumption of a perfect clustering scheme, implying the filter performance strongly depends on the performance of a clustering method. Popular clustering algorithms such as the K-means algorithm and expectation-maximization (EM) algorithm for GMMs have been applied to the PFGMM [84, 113].

The first PFGMM algorithm proposed in the literature, the Particle Gaussian Mixture Filter [84], utilizes K-means clustering which produces separated clusters for each Gaussian component of the GMM. When K is a fixed preassigned number of clusters, the K-means algorithm minimizes within-cluster-sum-of-squares (WCSS) (i.e., variance) by partitioning the data set into the K clusters. As a result, the covariance of each cluster generated by the K-means algorithm is as small as possible. A small covariance of each Gaussian mixture component is desirable outcome as the GSF is globally optimal only for linear systems and GMM distributions. In the presence of nonlinearities, typically, the smaller the covariance of a component, the smaller the nonlinear effects of the measurement function in the likely realizations of that component and hence the better the performance of the GSF.

Although it produces components with small covariance, the K-means algorithm does not guarantee an accurate GMM representation of the prior PDF. The EM algorithm for GMMs, produces a better approximation of the prior distribution by maximizing the likelihood function with respect to the clustering parameters, which are the means, covariance matrices, and weights of the components of a GMM. The K-means algorithm is a special case of the EM algorithm for GMMs in which all mixture weights are equal and all covariance matrices have spherical forms. The EM algorithm produces larger covariances than K-means, since it is not a hard clustering algorithm and allows for the components to overlap.

In this chapter, we propose two novel clustering algorithms to merge the benefits of K-means and EM and apply them to the PFGMM to improve its estimation performance. The two new types of clustering methods simultaneously minimize the covariance for each of the components of a GMM and maximize the likelihood function based on a fuzzy C-means (FCM) clustering algorithm [12]. The FCM algorithm is closely related to the K-means algorithm and EM algorithm for GMMs. In FCM clustering, the data points can belong to more than one cluster with different membership grades between 0 and 1. These membership grades represent the degree to which data points belong to the different clusters. The standard FCM algorithm, which is also referred to as soft K-Means algorithm, employs a weighting exponent on each fuzzy membership. The weighting exponent is also called the fuzzifier since it determines the level of the fuzziness of clustering. Unlike the approach of the standard FCM algorithm, the FCM algorithm can also be regarded as regularization of the K-means algorithm with a maximum entropy method [89].

Based on the new aspect of the FCM algorithm, Ref. [32] proposed the FCM with regularization by Kullback-Leibler information (KLFCM), which is similar to the EM algorithm for GMMs. Inspired from the standard FCM and KLFCM algorithm, in this chapter, we propose two types of clustering methods specifically designed to improve the performance of the PFGMM.

The remainder of this chapter is organized as follows. Sections 6.2 and 6.3 briefly describe the PFGMM and the FCM algorithm, respectively. Then, the new clustering algorithms are proposed in Section 6.4. Section 6.5 presents simulation results of the proposed algorithms, followed by some concluding remarks on the new methodologies and the results.

6.2 Particle Filter with Gaussian Mixture Models

Each algorithm's iteration starts from the knowledge of the prior distribution $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ which is approximated by N independent and identically distributed (i.i.d.) samples $\boldsymbol{x}_{k-1}^{(i)}$

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{6.1}$$

where k indicates the discrete time step, $\delta(\cdot)$ is the Dirac delta function, and $\boldsymbol{Y}_{k-1}$ is the set of all measurement vectors $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{k-1}\}$ where $\boldsymbol{y}_{k-1}$ is a measurement vector at the time step $k-1$. Following the same procedure as particle filters [7], a set of samples at the next time step is generated using the Markov transition kernel $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. The Markov kernel is defined by the dynamics of a system and the known statistics of the process noise.

The next step of the algorithm is to cluster the particles into Gaussian mixtures using a clustering algorithm such as the K-means algorithm or the EM algorithm for GMMs and the propagated distribution is then expressed as follows:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) \approx \sum_{j=1}^{K} \omega_{k|k-1}^{(j)} n(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) \tag{6.2}$$

where K is the predetermined number of clusters, $n(\boldsymbol{x}|\hat{\boldsymbol{x}}, P)$ represents the Gaussian PDF with mean $\hat{\boldsymbol{x}}$ and covariance P; and $\omega_{k|k-1}^{(j)}$, $\hat{\boldsymbol{x}}_{k|k-1}^{(j)}$, and $P_{k|k-1}^{(j)}$ are the weight, mean, and covariance matrix of the ith Gaussian component calculated by the K-means or EM clustering algorithm. The K-means and EM algorithm are similar in the sense that they use an iterative refinement approach to find the optimal clusters. The K-means is a hard clustering algorithm, which means each particle is associated uniquely with one cluster, as such it uses only the points in the same cluster to update each component's mean. The EM algorithm performs a soft assignment and approximates the PDF of $\boldsymbol{x}_k$ with several Gaussian components having different means, covariance matrices, and weights. The details of the K-means and EM algorithm are explained in [13].

Finally, we incorporate measurement information by updating the means, covariance matrices, and weights of all the components in the same way as the measurement update of the GSF. We then draw N i.i.d. samples from the GMM which is the posterior distribution. Since this is a recursive algorithm, we use the samples as a starting point for the next iteration. The details of the measurement update of the GSF and the algorithm to draw N i.i.d. samples from a GMM are explained in Chapter 3.

6.3 Fuzzy C-Means Clustering

The standard FCM and the KLFCM algorithm are reviewed in this section. In the standard FCM algorithm, the aim of the algorithm is to minimize the objective function which is defined as follows [12]:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} (u_{ij})^m d_{ij} \tag{6.3}$$

where C is the preassigned number of clusters and m is the weighting exponent for the fuzzification of memberships, $m \geq 1$. u_{ij} denotes the membership value of the i^{th} data sample for the j^{th} cluster and all membership values have the following constraints:

$$u_{ij} \geq 0, \forall i, j, \qquad \sum_{j=1}^{C} u_{ij} = 1 \tag{6.4}$$

The dissimilarity function d_{ij} which is a measure of the Euclidean distance between i^{th} data sample $\boldsymbol{x}^{(i)}$ and the center of the j^{th} cluster $\boldsymbol{\mu}_j$ is expressed as follows:

$$d_{ij} = \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right)^{\mathrm{T}} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right) \tag{6.5}$$

The optimal values of the clustering parameters u_{ij} and $\boldsymbol{\mu}_j$ are calculated using a fixed-point iteration scheme, which is similar to the EM algorithm for GMMs. The algorithm is summarized as follows:

1. Initialize the membership values u_{ij} and set the objective function value to infinity.

2. Calculate the cluster centers $\boldsymbol{\mu}_j$ using the current membership values.

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} (u_{ij})^m \boldsymbol{x}^{(i)}}{\sum_{i=1}^{N} (u_{ij})^m} \tag{6.6}$$

3. Estimate the new membership values u_{ij} using the current cluster centers to minimize the objective function.

$$u_{ij} = \left[\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{6.7}$$

4. Evaluate the objective function and check for convergence of it. If the convergence criterion is not satisfied, replace the old membership values with the new ones and return to step 2.

After a number of iterations, the clustering parameters are optimized to minimize the objective function.

Fuzziness is the level of overlap between clusters (more overlap equals less defined or fuzzier boundaries). In the standard FCM algorithm the weighting exponent m determines the level of the fuzziness of clustering, so it is also called fuzzifier [12]. In other words, the partition is getting fuzzier as the fuzzifier has a larger value. On the other hand, the memberships u_{ij} converges to 0 or 1 when the fuzzifier m has the minimum value of one, which means the FCM algorithm reduces exactly to the K-means algorithm.

In addition, the FCM algorithm is closely related to the EM algorithm for GMMs. Ref. [31] defined a fuzzy covariance matrix for the FCM algorithm so that different clusters can have different geometric shapes in the clustering. Moreover, Ref. [89] shows that the FCM algorithm can be regraded as the regularization of the K-means algorithm with a maximum entropy method, and Ref. [32] demonstrates that the EM algorithm of the GMMs can be casted as a penalized version of the hard means clustering algorithm. As a result, the FCM clustering with the regularizer by

Kullback-Leibler (KL) information, called KLFCM, can be made the same algorithm as the EM algorithm for GMMs [35]. The objective function of the KLFCM is expressed as follows:

$$\begin{aligned} J_{KL} = & \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} d'_{ij} + \lambda \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log \frac{u_{ij}}{\pi_j} + \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log |\Sigma_j| \\ & - \sum_{i=1}^{N} \boldsymbol{\eta}_i \left[\sum_{j=1}^{C} u_{ij} - 1 \right] - \tau \left[\sum_{j=1}^{C} \pi_j - 1 \right] \end{aligned} \tag{6.8}$$

where the dissimilarity function d'_{ij} of the KLFCM is the Mahalanobis distance,

$$d'_{ij} = \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right)^{\mathrm{T}} \Sigma_j^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right) \tag{6.9}$$

and λ is the fuzzifier, $\lambda \geq 0$. Σ_j and π_j are the covariance matrix and weight of the j^{th} cluster, and $\boldsymbol{\eta}_i$ and τ are Lagrangian multipliers whose corresponding terms respectively indicate the constraints on membership values and weights.

The necessary conditions for optimality of (6.8) are derived as follows:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} u_{ij} \boldsymbol{x}^{(i)}}{\sum_{i=1}^{N} u_{ij}} \tag{6.10}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} u_{ij} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{N} u_{ij}} \tag{6.11}$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} u_{ij} \tag{6.12}$$

where

$$u_{ij} = \frac{\pi_j \exp\left(-\frac{1}{\lambda} d'_{ij}\right) |\Sigma_j|^{-1/\lambda}}{\sum_{k=1}^{C} \pi_k \exp\left(-\frac{1}{\lambda} d'_{ik}\right) |\Sigma_k|^{-1/\lambda}} \tag{6.13}$$

The iteration rule is equivalent to the standard FCM algorithm. In the KLFCM algorithm, the fuzzifier λ tunes the degree of fuzziness of the membership values.

For instance, all memberships are getting closer to $1/C$ as we are putting more weight on the relative entropy term. Thus, the larger the fuzzifier, the fuzzier the memberships. The KLFCM algorithm is the same as the EM algorithm for GMMs when the fuzzifier λ is equal to 2.

6.4 Clustering Methods for PFGMM

This section presents two different clustering methods to simultaneously minimize the covariance for each of the components of a GMM and maximize the likelihood function for the PFGMM. Both these features are key contributors to PFGMM estimation performance.

6.4.1 KLFCM with Weighting Exponent

The objective function of the KLFCM with the fuzzification coefficient 2 is expressed as follows:

$$\begin{aligned} J_{KL2} = & \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}\log u_{ij} \\ & + \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}\left[\log\frac{1}{\pi_j} - \log\frac{(2\pi)^{-d_x/2}}{(2\pi)^{-d_x/2}|\Sigma_j|^{-1/2}}\exp\left(-\frac{1}{2}d'_{ij}\right)\right] \\ & - \sum_{i=1}^{N}\boldsymbol{\eta}_i\left[\sum_{j=1}^{C} u_{ij} - 1\right] - \tau\left[\sum_{j=1}^{C}\pi_j - 1\right] \end{aligned} \tag{6.14}$$

which is equivalent to the following function:

$$J_{KL2} = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}\log u_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} d''_{ij} - \sum_{i=1}^{N} \boldsymbol{\eta}_i \left[\sum_{j=1}^{C} u_{ij} - 1\right] - \tau\left[\sum_{j=1}^{C}\pi_j - 1\right] \tag{6.15}$$

where d_x is the dimension of the state $\boldsymbol{x}$ and the dissimilarity function d''_{ij} is defined as follows:

$$d''_{ij} = \log\left(\frac{1}{(2\pi)^{-d_x/2}\pi_j n(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}_j, \Sigma_j)}\right) \tag{6.16}$$

The first term (i.e., entropy term) of the objective function does not have the fuzzifier λ because it is set to be equal to 2. Therefore, the modified KLFCM algorithm with the objective function could only act as the EM algorithm. To assign the hard clustering property of the K-means to the modified algorithm, we make it include a new fuzzifier. If the new fuzzifier λ' is added to the first term, it has the same form as the FCM algorithm with a maximum entropy method, and the updating rule of the membership values are expressed as follows [89]:

$$u_{ij} = \frac{\exp\left(-\frac{1}{\lambda'}d''_{ij}\right)}{\sum_{k=1}^{C}\exp\left(-\frac{1}{\lambda'}d''_{ik}\right)} = \frac{\pi_j^{-1/\lambda'}\exp\left(-\frac{1}{2\lambda'}d'_{ij}\right)|\Sigma_j|^{-1/2\lambda'}}{\sum_{k=1}^{C}\pi_k^{-1/\lambda'}\exp\left(-\frac{1}{2\lambda'}d'_{ik}\right)|\Sigma_k|^{-1/2\lambda'}} \tag{6.17}$$

Although the memberships become less fuzzier as the fuzzifier λ' is smaller, it simultaneously loses the property of the EM algorithm since the fuzzifier changes and distorts the Gaussian distributions. Consequently, the new fuzzifier makes the modified algorithm very similar to the KLFCM algorithm. To cope with the problem, we can change the objective function into the standard FCM form by removing the first term and introducing the weighting exponent ($m \geq 1$) on the membership

values. Then, the modified objective function is expressed as follows:

$$J_{KL2'} = \sum_{i=1}^{N}\sum_{j=1}^{C}(u_{ij})^m d''_{ij} - \sum_{i=1}^{N}\boldsymbol{\eta}\left[\sum_{j=1}^{C}u_{ij} - 1\right] - \tau\left[\sum_{j=1}^{C}\pi_j - 1\right] \tag{6.18}$$

The updating rules for the means, covariance matrices, weights of all clusters and memberships are then derived as follows:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N}(u_{ij})^m \boldsymbol{x}^{(i)}}{\sum_{i=1}^{N}(u_{ij})^m} \tag{6.19}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N}(u_{ij})^m (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{N}(u_{ij})^m} \tag{6.20}$$

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N}(u_{ij})^m \tag{6.21}$$

and

$$u_{ij} = \left[\sum_{k=1}^{C}\left(\frac{d''_{ij}}{d''_{ik}}\right)^{\frac{1}{m-1}}\right]^{-1} \tag{6.22}$$

The proposed algorithm maximizes the log likelihood function, which is the same as the EM algorithm for GMMs. Moreover, the memberships are proportional to the exponential of the log likelihood function of the GMM. In other words, even if the weighting exponent m is changed to tune the level of the fuzziness of the clustering, the form of the log likelihood function (i.e., GMM) in the memberships is maintained. Therefore, the degree of fuzziness of the memberships can be determined by the weighting exponent m with retaining the property of the EM algorithm. As in the standard FCM algorithm, the partition becomes more distinct as the fuzzifier m has a smaller value. When m is set to be very close to the minimum value of 1, however, most of the memberships converges to 0 or 1, thus making

some weights of the clusters zero [12]. As a result, the covariance for the rest of the components of a GMM is increased. To prevent this problem, a regularization term with a positive parameter κ is included in the objective function as follows:

$$\begin{aligned} J_{mKL} = & \sum_{i=1}^{N}\sum_{j=1}^{C}(u_{ij})^m d''_{ij} - \kappa \sum_{i=1}^{N}\sum_{j=1}^{C}\log(\pi_j) \\ & - \sum_{i=1}^{N}\boldsymbol{\eta}\left[\sum_{j=1}^{C}u_{ij} - 1\right] - \tau\left[\sum_{j=1}^{C}\pi_j - 1\right] \end{aligned} \tag{6.23}$$

By introducing this new regularization term, all the weights of the clusters are getting closer to $1/C$ as the value of κ is lager because the sum of π_j with respect to C is constrained to be 1. The updating rule for the weights of the clusters is then changed into

$$\pi_j = \frac{2\sum_{i=1}^{N}(u_{ij})^m + \kappa N}{2\sum_{i=1}^{N}\sum_{j=1}^{C}(u_{ij})^m + \kappa NC} \tag{6.24}$$

and the others (6.19), (6.20), and (6.22) remain the same. The proposed algorithm is named the KLFCM clustering with weighting exponent (mKLFCM).

6.4.2 KLFCM with Ridge Regularization

Another proposed clustering method is to add a new regularization term to the objective function of the KLFCM with the fuzzification coefficient 2. Regularization is one of the most important concepts in machine learning (ML) and the most well known regularization techniques are ridge and least absolute shrinkage and selection operator (LASSO) regularization: the two regularization techniques are used to reduce the magnitude of irrelevant coefficients of a model and avoid overfitting [73]. As done in ridge regularization, a new regularization term is added

to (6.8) with the fuzzification coefficient 2 in order to restrict the determinant of the covariance of each cluster to the determinant of the total covariance divided by the number of total clusters as follows:

$$\begin{aligned} J_{RKL'} = & \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} d'_{ij} + \alpha \sum_{i=1}^{N}\sum_{j=1}^{C} \log\left(|\Sigma_j| - \frac{|\Sigma_T|}{C}\right)^2 \\ & + 2\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log \frac{u_{ij}}{\pi_j} + \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log |\Sigma_j| \\ & - \sum_{i=1}^{N} \boldsymbol{\eta}_i \left[\sum_{j=1}^{C} u_{ij} - 1\right] - \tau \left[\sum_{j=1}^{C} \pi_j - 1\right] \end{aligned} \tag{6.25}$$

where α is the penalty parameter and Σ_T is the covariance matrix of the total samples. To put it another way, the new clustering algorithm minimizes the following objective function:

$$J_{KL2} \quad \text{subject to} \quad \sum_{j=1}^{C} \log\left(|\Sigma_j| - \frac{|\Sigma_T|}{C}\right)^2 < R_c \tag{6.26}$$

where J_{KL2} is given by (6.15) and R_c is a parameter. Figure 6.1 shows an example of how the new regularization term in the objective function can be geometrically interpreted when the number of clusters is 2. If the new regularization term does not exist (i.e. $\alpha = 0$), which means there is no inequality constraint, the center of the ellipse will be the optimal value of the determinant of the each cluster's covariance. Figure 6.1 shows, however, the optimal values for the covariance matrices of the objective functions in (6.25) and (6.26) are given by the first point at which the ellipse (blue) contacts the constraint region (orange) due to the new regularizer (the inequality constraint). As a result, the large value of α (the smaller value of R_c)

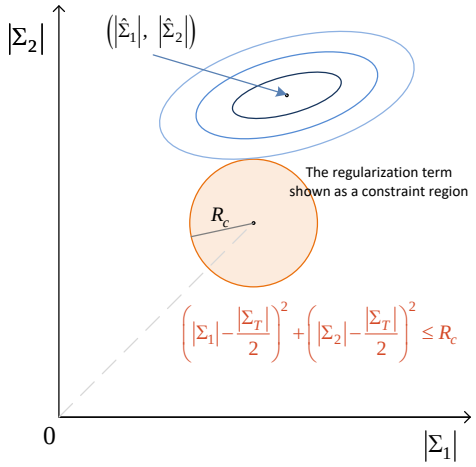


Figure 6.1: Geometric interpretation of the new regularizer

will shrink the determinants of all the covariance matrices towards the same value $(|\Sigma_T|/C)$.

The updating rules for the clustering parameters of (6.25) are the same as those of (6.8) with $\lambda = 2$ except for the covariance matrices. For the new objective function (6.25), the updating rule for the covariance matrices is not fixed because it varies depending on the dimension of state. Therefore, to find the consistent closed-form solution for the updating rule of the covariance matrices, we substitute Σ_j with $\beta_j M_j$ where $|M_j| = 1$, $\beta_j = |\Sigma_j|^{1/d_x}$. Then, the objective function (6.25) can be

changed into the following equivalent objective function:

$$\begin{aligned} J_{RKL} = & \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} d'_{ij} + \alpha \sum_{i=1}^{N}\sum_{j=1}^{C} \left(\frac{D}{\beta_j} - 1\right)^2 \\ & + 2\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log \frac{u_{ij}}{\pi_j} + \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \log |\Sigma_j| \\ & - \sum_{i=1}^{N} \boldsymbol{\eta}_i \left[\sum_{j=1}^{C} u_{ij} - 1\right] - \tau \left[\sum_{j=1}^{C} \pi_j - 1\right] - \sum_{j=1}^{C} \xi_j \log |M_j| \end{aligned} \tag{6.27}$$

where $\xi_1, \cdots, \xi_C$ are Lagrangian multipliers and

$$D = \left(\frac{|\Sigma_T|}{C}\right)^{1/d_x} \tag{6.28}$$

Then, the updating rules for the M_j and β_j are derived as follows:

$$M_j = \frac{\sum_{i=1}^{N} u_{ij} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T}{|\sum_{i=1}^{N} u_{ij} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T|^{1/d_x}} \tag{6.29}$$

$$\beta_j = \frac{1}{2d_x \sum_{i=1}^{N} u_{ij}} \left[A - B + \sqrt{(A-B)^2 + C}\right] \tag{6.30}$$

where

$$A = \sum_{i=1}^{N} u_{ij} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right)^{\mathrm{T}} M_j^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\right) \tag{6.31}$$

$$B = 2\alpha N D \tag{6.32}$$

and

$$C = 8\alpha d_x N D^2 \sum_{i=1}^{N} u_{ij} \tag{6.33}$$

The proposed algorithm is named the KLFCM clustering with ridge regularization (RKLFCM).

6.4.3 Analysis of the Proposed Clustering Methods

In the mKLFCM clustering algorithm, it is crucial to select the most appropriate weighting exponent m which determines the performance of the PFGMM. It is desirable for the PFGMM to have an accurate GMM representation of the prior PDF while each component has a small enough covariance matrix such that nonlinear measurements can be accurately approximated by linearization in the support of each component. In this section, we investigate the impact of the weighting exponent on the clustering algorithm and find the optimal value for the PFGMM based on the biggest determinant of the covariance matrix of the components and the Jensen-Shannon distance (JSD). Note that the smaller the biggest determinant of the covariance matrix, the better the performance of the PFGMM. The JSD, which is a metric, is used to measure the similarity between two probability distributions and given by

$$J(p_x||q_x) = \sqrt{\frac{1}{2}D_{KL}(p_x||m_x) + \frac{1}{2}D_{KL}(q_x||m_x)} \tag{6.34}$$

where

$$D_{KL}(p_x||q_x) = \sum_{\boldsymbol{x}\in S} p_x(\boldsymbol{x}) \log\left(\frac{p_x(\boldsymbol{x})}{q_x(\boldsymbol{x})}\right) \tag{6.35}$$

$$m_x = \frac{1}{2}(p_x + q_x) \tag{6.36}$$

and S is the support for $\boldsymbol{x}$, and p_x and q_x are two different probability distributions. As the difference between the two distributions becomes smaller, the JSD also gets smaller.

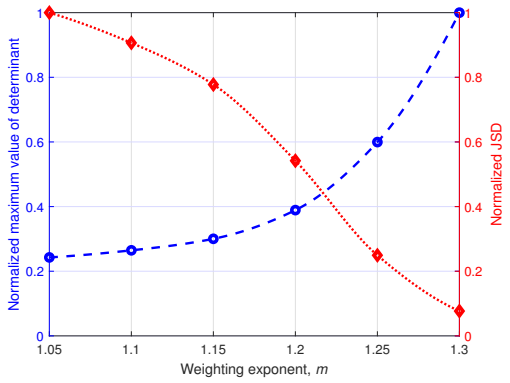


Figure 6.2: Analysis of the weighing exponent in the mKLFCM

To analyze the impact of the weighting exponent on the clustering algorithm, a Monte Carlo analysis is performed with 100 simulations. For each simulation, 10,000 particles are drawn from the two-dimensional normal distribution. The mKLFCM uses 3 clusters and the parameter κ is set to be 0.05. Figure 6.2 shows the normalized maximum value of determinant and JSD according to the weighting exponent and the original values are listed in Table 6.1. As mentioned above, the greater the weighting exponent is, the lower the JSD value becomes. However, the maximum value of the determinant of covariance also tends to increase with the weighting exponent. Figure 6.2 shows that the optimal value of the weighting exponent is 1.216 when samples are from a normal distribution. In addition, Table 6.1 implies that as the weighting exponent is larger, the maximum value of determinant approaches to 1, which is the determinant of the sample covariance. Overall, the mKLFCM behaves like the K-means algorithm as the weighting exponent decreases, whereas it functions like the EM algorithm as the weighting exponent increases.

Table 6.1: Analysis of the weighing exponent in the mKLFCM

	1.05	1.10	1.15	1.20	1.25	1.30	1.35
Maximum determinant	0.2302	0.2510	0.2848	0.3690	0.5686	0.9484	0.9737
JSD	1.8440	1.6716	1.4339	0.9991	0.4598	0.1415	0.1318

Table 6.2: Analysis of the penalty parameter in the RKLFCM

	0.0001	0.001	0.01	0.1	1	10	100
Maximum determinant	0.7774	0.7714	0.7179	0.5032	0.3567	0.3355	0.3334
JSD	0.2338	0.2365	0.2654	0.4917	0.8018	0.8691	0.8766

A key element of the RKLFCM is how to determine the penalty parameter α. A Monte Carlo analysis with the same simulation conditions as above is performed. Figure 6.3 presents the normalized maximum value of determinant and JSD according to the penalty parameter and the original values are illustrated in Table 6.2. As the penalty parameter increases, the maximum value of determinant decreases. Table 6.2 shows that the maximum value of determinant approaches to 1/3 which is ratio of the sample covariance to the number of clusters used for the algorithm. On the other hand, the JSD value tends to increase with the penalty parameter. As can be seen in Figure 6.3, 0.1414 is the optimal value of the parameter when samples are drawn from a normal distribution. To summarize: the RKLFCM becomes similar to the EM algorithm as the weighting exponent decreases, whereas it operates in a similar way of the K-means algorithm as the weighting exponent increases.

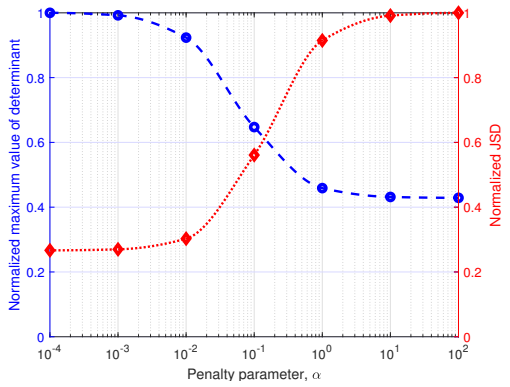


Figure 6.3: Analysis of the penalty parameter in the RKLFCM

6.5 Numerical Results

To evaluate the clustering algorithms proposed in this chapter, two different examples are considered: a simple motivating example (used in Refs. [103] and [117]) and the vector nonstationary growth model (used in Refs. [70] and [43]).

6.5.1 Single Step Example

Consider the following simple motivating example. An initial bivariate normal random vector $\boldsymbol{x}_0$ is distributed as

$$\boldsymbol{x}_0 \sim n\left(\boldsymbol{x}_0; \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}\right) \tag{6.37}$$

and a range measurement of 1.5 with measurement noise variance of 0.01 is available. Figure 6.4 shows contour plots of the given prior and likelihood distribution, and Figure 6.5 presents a contour plot of the true posterior distribution. The measurement is nonlinear so that it causes the banana-shaped posterior distribution which is cannot be accurately estimated by linear estimators such as EKF and UKF

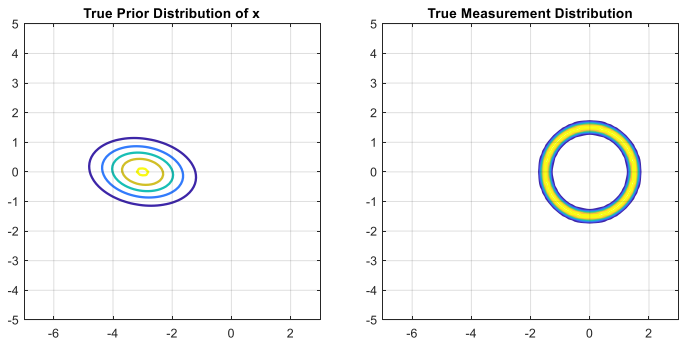


Figure 6.4: Prior and measurement distribution for example 1

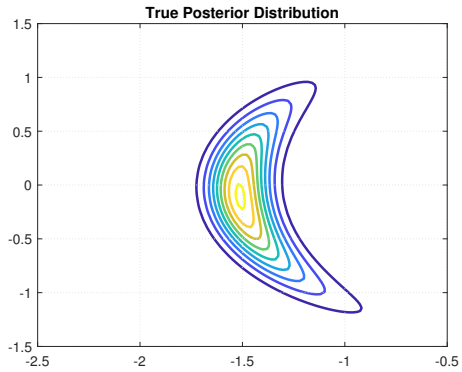


Figure 6.5: True posterior distribution for example 1

[103, 117].

In this example, we compare the clustering performances of the four different clustering methods; the K-means algorithm, the EM algorithm for GMMs, and the two algorithms proposed here [mKLFCM and RKLFCM]. We test the clustering algorithms to see whether they approximate (a) the prior distribution by a GMM, (b) the nonlinear measurement function by a linear function for the measurement update of each component, and (c) the true posterior distribution by an updated

GMM.

For Case (a) and (c), the JSD is used to measure the distance between the true distribution and approximated distribution by a GMM. Moreover, for each Gaussian component, the approximated posterior distribution of the EKF solution is compared to that of the second-order EKF (SEKF) solution using the JSD for Case (b) in order to check whether the linearization of the measurement function is valid in a region of the component [103]. If the covariance of a component is small enough such that the nonlinearities of the measurement function are negligible, then the difference between the distributions will be small. For Case (b), the weighted sum of the JSD values is used for a GMM with the weights of all the components.

All of the clustering algorithms use 30 clusters with 10,000 particles. The proposed clustering algorithms use the following tuning parameters: $m = 1.216$ and $\kappa = 0.05$ for the mKLFCM, and $\alpha = 0.1414$ for the RKLFCM. For the EM, mKLFCM, and RKLFCM algorithm, the initial clustering parameters are selected using the K-means algorithm [13]. A Monte Carlo analysis is performed with 100 simulations and the averaged JSD values for all of the cases are listed in Table 6.3. For Case (a), the EM algorithm for GMMs and RKLFCM algorithm are comparable, while the K-Means algorithm results in the worst performance among the four clustering algorithms. For Case (b), however, the K-means algorithm is better than any of other algorithms and the EM algorithm gives the worst performance. Finally, Table 6.3 shows that, for Case (c), the proposed algorithms [mKLFCM and RKLFCM] outperform the K-means and EM algorithm for GMMs, which means the proposed algorithms are more suitable for the PFGMM.

Table 6.3: Monte Carlo averaged JSD for Exemple 1 (100 simulations)

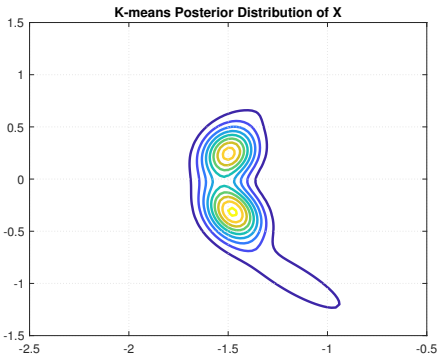
	K-means	EM	mKLFCM	RKLFCM
Case (a)	2.6467	0.5866	1.5423	0.7044
Case (b)	2.3689	5.8114	3.4402	4.5364
Case (c)	6.2189	6.0616	5.2056	5.0687

Moreover, for Case (c), Figure 6.6 displays the PFGMM solutions with the K-means, EM for GMMS, mKLFCM, and RKLFCM clustering algorithms for a single simulation. Note that the PFGMM solutions with the four clustering methods succeed in capturing the curvature shape of the true posterior distribution. If a linear estimator of nonlinear systems such as the EKF or UKF is used for this example, the curvature shape of the true distribution cannot be replicated by the first two moments of the estimator. This figure also shows that the PFGMM solutions with the proposed clustering methods can more accurately adapt to nonlinearities of the measurement function than the PFGMM with the K-means and EM algorithm for GMMs.

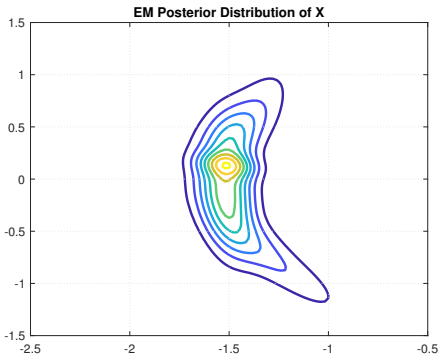
6.5.2 Vector Nonstationary Growth Model

Consider the discrete-time highly nonlinear bivariate dynamic system and measurement model given by [43, 70]

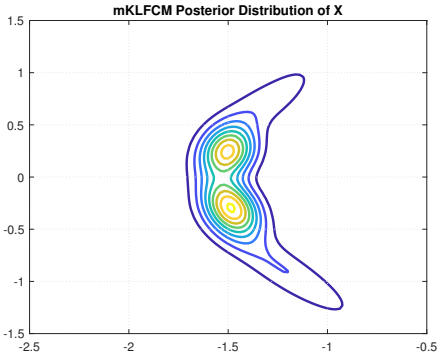
$$\begin{aligned} x_{k+1,1} &= \frac{x_{k,1}}{2} + 25\frac{x_{k,1}}{1+x_{k,1}^2} + 8\cos(1.2k) + \nu_{k,1} \\ x_{k+1,2} &= \frac{x_{k,2}}{2} + 25\frac{x_{k,2}}{1+x_{k,2}^2} + 8\cos(1.2k) + \nu_{k,2} \end{aligned} \tag{6.38}$$



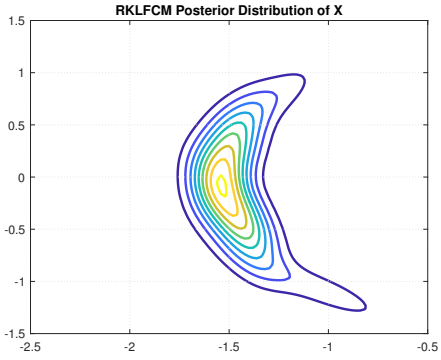
(a) PFGMM with the K-means algorithm Posterior



(b) PFGMM with the EM algorithm for GMMs Posterior



(c) PFGMM with the mKLFCM algorithm Posterior



(d) PFGMM with the RKLFCM algorithm Posterior

Figure 6.6: The PFGMM solutions with the four different clustering algorithms for a single simulation are presented with (a) the JSD value (the K-means) = 5.6130, (b) the JSD value (the EM algorithm for GMMs) = 5.4898, (c) the JSD value (the mKLFCM) = 4.4212, and (d) the JSD value (the RKLFCM) = 4.2193.

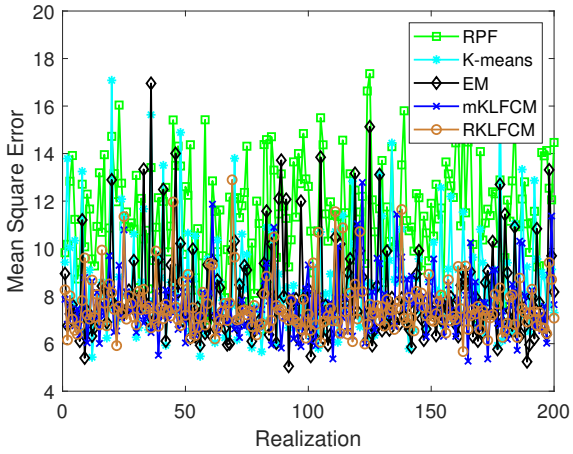
$$\begin{aligned} y_{k+1,1} &= \frac{x_{k+1,1}^2 + x_{k+1,2}^2}{20} + \eta_{k+1,1} \\ y_{k+1,2} &= \frac{x_{k+1,1}^2 - x_{k+1,2}^2}{10} + \eta_{k+1,2} \end{aligned} \tag{6.39}$$

where the process noise $\boldsymbol{\nu}_k = [\nu_{k,1}, \nu_{k,2}]^{\mathrm{T}}$ and the measurement noise $\boldsymbol{\eta}_{k+1} = [\eta_{k+1,1}, \eta_{k+1,2}]^{\mathrm{T}}$ are assumed to be independent zero-mean Gaussian random vari-

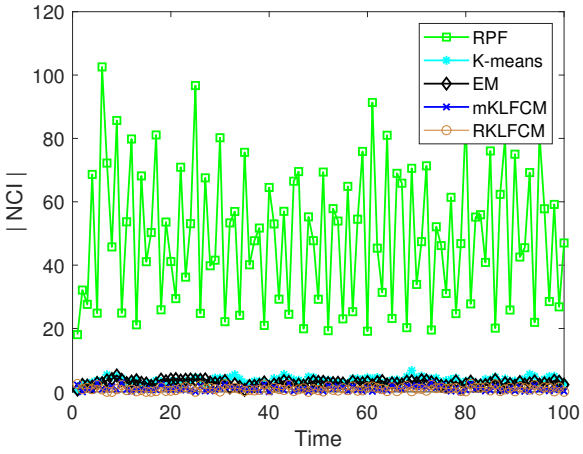
ables with variances of $Q = 10I_{2\times2}$ and $R = I_{2\times2}$, respectively, and $I_{2\times2}$ is a 2×2 identity matrix.

The model is highly nonlinear and the cosine term in the dynamic equation varies with time k. In this example, a Monte Carlo analysis is performed with 200 simulations, and each simulation has a time span of $k = [0, 100]$. The estimation performances of the PFGMM with four different clustering algorithms and the regularized particle filter (RPF) [7] are compared based on the root-means-square error (RMSE) and the noncredibility index (NCI) [55]. The RMSE for each Monte Carlo simulation is computed from the true and estimated states at each time k. The NCI measures the difference between the ideal error covariance matrix and the estimated error covariance matrix. The NCI metric is a balanced measure of the consistency of the estimators. When the difference between the ensemble error covariance matrix and the filter's error covariance matrix is small, the NCI value should be zero or nealy zero at all times [55]. A total of 200 particles is used in all of the algorithms and the UKF measurement update for 3 clusters is used for the PFGMM. The tuning parameters for the proposed algorithms and selecting method for initial clustering parameters are the same as those used in the example 1.

Figure 6.7 shows the RMSE and the absolute NCI value of the 200 simulations. The time-averaged value of the RMSE and NCI of the five algorithms are described in Table 6.4. The proposed algorithms outperform the K-means and EM algorithm for GMMs in terms of accuracy and consistency. Moreover, the figure shows that the PFGMMs perform better than the RPF both in the accuracy and consistency. The computation time for filtering run in MATLAB on a 3.2 GHz



(a) RMSE



(b) | NCI |

Figure 6.7: The average RMSE and the absolute NCI value for 200 Monte Carlo simulations

hexa-core Windows operation systems is also presented in Table 6.4. In terms of computation time, the proposed algorithms are cheaper than the EM algorithm for GMMs and RPF while the best performance is produced with the K-means algorithm.

Table 6.4: Monte Carlo averaged RMSE, NCI, and computation time for Example 2 (200 simulations)

	RMSE	NCI	Computation time
RPF	11.7031	49.4944	1.4342
K-Means	8.7378	3.6591	0.4889
EM	8.2914	2.7653	1.0130
mKLFCM	7.6699	1.0270	0.7232
RKLFCM	7.6417	0.9482	0.6525

6.6 Chapter's Summary

In this chapter, two new clustering algorithms are proposed whose performance index minimizes the covariance of each of the components of a Gaussian mixture model and maximizes the likelihood function simultaneously. The two new clustering algorithms are based on fuzzy C-means with regularization by Kullback-Leibler information. The objective function of the first method has the same form of the standard fuzzy C-means algorithm with an added weighting exponent. In the second method, a new regularization term is included in the objective function of the baseline algorithm. As a result, both of the proposed clustering algorithms compensate for the drawbacks of the K-means and expectation-maximization algorithm for the particle filter with Gaussian mixture models. Two numerical examples show that the particle filter with Gaussian mixture models with the proposed clustering algorithms provide better performance than the particle filter with Gaussian mixture models with the K-means or expectation-maximization algorithm as well as the regularized particle filter in terms of the accuracy and consistency.

Chapter 7

Conclusions

In this dissertation, a study of new Bayesian approaches combining the benefits of the Gaussian sum filter and particle filter was performed. The filtering techniques introduced in Chapters 3 and 4 of this dissertation develop methods in which each and every particle of particle filter is made into a Gaussian mixture component, either using the properties of Dirac delta function or using kernel density estimation. The filtering algorithms presented in Chapters 5 and 6 entail clustering methods to group several particles into each Gaussian component.

The first contribution of this dissertation lies in the derivation of a new sequential Monte Carlo algorithm that samples from a Gaussian mixture model approximation of the posterior distribution. Each sample of the prior distribution is treated as a Gaussian component with a collapsed zero covariance matrix and the Gaussian sum filter is used to calculate the posterior distribution. Two small modifications of the baseline algorithm were also developed to improve the accuracy. Compared to the bootstrap particle filter and regularized particle filter, the performances in terms of accuracy, consistency, and mean computation time of the proposed algorithms were analyzed with four different examples using Monte Carlo simulations.

The second contribution of this work is the introduction of the modified kernel-based ensemble Gaussian mixture filtering for orbit determination with sparse measurement data. Silverman's rule of thumb is used for bandwidth estimation to efficiently track SOs in LEO with short and sparse observation data. Moreover, to improve the filter's accuracy and consistency, the proposed algorithm is applied in conjunction with the use of equinoctial orbital elements, and simulations results showed the effectiveness of the modified algorithm compared to the unscented Kalman filter and the state-of-the-art Gaussian sum filter, AEGIS.

Future work on the first and second subjects should include adapting the number of needed particles to a problem. The proposed filters are based on sequential Monte Carlo methods whose performances in terms of accuracy and consistency rely heavily on the number of particles used in the algorithms. Increasing the number of particles, however, results in a high computational cost. Therefore, the development of an adaptive algorithm with an appropriate metric would make the proposed algorithms more accurate and robust as well as faster.

The third and fourth contributions of this study reside in the introduction of the particle filter with Gaussian mixture models, a new sequential Monte Carlo algorithm based on a clustering algorithm. This new approach uses a clustering algorithm to combine particle filters and Gaussian sum filter; the propagated particles are clustered to recover a Gaussian mixture model representation of the prior probability density function. The expectation-maximization clustering algorithm for Gaussian mixture models was applied to deal with a challenging problem in the context of simultaneous space object tracing and characterization. Moreover, two

new clustering algorithms for the particle filter with Gaussian mixture models were developed to merge the benefits of the K-means and expectation-maximization clustering algorithms. Numerical simulations results demonstrated that the proposed algorithms are efficient.

Future work on the third and fourth subjects should include reducing the computational burden of the proposed algorithms. The appropriate number of clusters and particles could be adaptively selected according to a given problem. Before developing a metric for the adaptive algorithm, the relationship between the performances of clustering algorithms and the number of particles need to be precisely analyzed.

Bibliography

[1] The European Space Agency. Space debris by the numbers. https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers, 2020.

[2] D. Alspach and H. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, August 1972. doi:10.1109/SAP.1970.270017.

[3] Jeffrey L. Anderson and Stephen L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999. doi:10.1175/1520-0493(1999)127¡2741:AMCIOT¿2.0.CO;2.

[4] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[5] Ienkaran Arasaratnam and Simon Haykin. Cubature kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254 – 1269, June 2009. doi:10.1109/TAC.2009.2019800.

[6] Ienkaran Arasaratnam, Simon Haykin, and Robert J. Elliott. Discrete-time nonlinear filtering algorithms using gauss-hermite quadrature. *Proceedings*

of the IEEE, 95(5):953–977, 2007.
doi:10.1109/JPROC.2007.894705.

[7] M.S. Arulampalam, S. Maskell, N Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
doi:10.1109/78.978374.

[8] Sanjeev Arulampalam, Neil Gordon, and Branko Ristic. *Beyond the Kalman Filter: Particle Filters for Tracking Applications Partical Filters for Tracking Applications*. Artech House, 2004.
ISBN:978-1-5805-3631-8.

[9] Michael Ashikhmin and Peter Shirley. An anisotropic phong light reflection model. *Journal of Graphics Tools*, 5(2):25–32, 2000.
doi:10.1080/10867651.2000.10487522.

[10] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2001.

[11] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[12] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, NY, 1981.

[13] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, NY, 2006. pp. 430-441.

[14] R. A. Broucke and P. J. Cefola. On the equinoctial orbit elements. *Celestial Mechanics*, 5(3):303–310, 1972.
doi:10.1007/BF01228432.

[15] Richard S. Bucy and Kenneth D Senne. Digital synthesis of non-linear filters. *Automatica*, 7(3):287–298, 1971.
doi:10.1016/0005-1098(71)90121-X.

[16] Thibaut Castaings, Benjamin Pannetier, Florent Muller, and Michele Rombaut. Track initiation in low-earth-orbit objects using statistical modeling of sparse observations. *IEEE Transactions on Aerospace and Electronic Systems*, 51(1):258–269, 2015.
doi:10.1109/TAES.2014.130484.

[17] Ryan D. Coder, Marcus J. Holzinger, and Richard Linares. Three-degree-of-freedom estimation of agile space objects using marginalized particle filters. *Journal of Guidance, Control, and Dynamics*, 41(2):388–400, 2018.
doi:10.2514/1.G001980.

[18] Ryan D. Coder, Charles J. Wetterer, Kris M. Hamada, Marcus J. Holzinger, and Moriba K. Jah. Inferring active control mode of the hubble space telescope using unresolved imagery. *Journal of Guidance, Control, and Dynamics*, 41(1):164–170, 2018.
doi:10.2514/1.G002223.

[19] Kyle J. DeMars, Robert H. Bishop, and Moriba K. Jah. An entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 36(4):1047–1057, July–August' 2013.
doi:10.2514/1.58987.

[20] Kyle J. DeMars, Moriba K. Jah, and Paul W. Schumacher. Initial orbit determination using short-arc angle and angle rate data. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2628–2637, 2012.
doi:10.1109/TAES.2012.6237613.

[21] Andrew D. Dianetti, Ryan Weisman, and John L. Crassidis. Observability analysis for improved space object characterization. *Journal of Guidance, Control, and Dynamics*, 41(1):137–148, 2018.
doi:10.2514/1.G002229.

[22] J. R. Dormand and R. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, March 1980.

[23] A. Doucet, N. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.
ISBN:978-1-4757-3437-9.

[24] Siwei Fan and Carolin Frueh. A direct light curve inversion scheme in the presence of measurement noise. *The Journal of the Astronautical Sciences*,

pages 1–22, 2019.
doi:10.1007/s40295-019-00190-3.

[25] Friedrich Faubel and Dietrich Klakow. Further improvement of the adaptive level of detail transform: Splitting in direction of the nonlinearity. In *Signal Processing Conference, 2010 18th European*, pages 850–854. IEEE, 2010. ISSN:2219-5491.

[26] Friedrich Faubel, John McDonough, and Dietrich Klakow. The split and merge unscented gaussian mixture filter. *IEEE Signal Processing Letters*, 16(9):786–789, 2009.
doi:10.1109/LSP.2009.2024859.

[27] William M. Folkner, James G. Williams, Dale H. Boggs, Ryan S. Park, and Petr Kuchynka. The planetary and lunar ephemerides DE430 and DE431. IPN Progress Report 42-196, Jet Propulsion Laboratory, California Institute of Technology, http://ipnpr.jpl.nasa.gov/progress_report/42-196/196C.pdf, February 2009.

[28] Luc Fraiture. The Information Dilution Theorem. *ESA Journal*, 10:381–386, 1986.

[29] Arthur Gelb, editor. *Applied Optimal Estimation*. The MIT press, Cambridge, MA, 1974.

[30] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Pro-*

cessing, IEE Proceedings F, 140(2):107–113, 1993.
doi:10.1049/ip-f-2.1993.0015.

[31] Donald E. Gustafson and William C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pages 761–766, Jan. 1979.
doi:10.1109/CDC.1978.268028.

[32] Richard J. Hathaway. Another interpretation of the em algorithm for mixture distributions. *Statistics and Probability Letters*, 4(2):53–56, 1986.
doi:10.1016/0167-7152(86)90016-7.

[33] Frank Havlak and Mark Campbell. Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. *IEEE Transactions on Robotics*, 30(2):461–474, 2014.
doi:10.1109/TRO.2013.2291620.

[34] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.
doi:10.1177/0278364913509675.

[35] Katsuhiro Honda Hidetomo Ichihashi, Kiyotaka Miyagishi. Fuzzy c-means clustering with regularization by k-l information. In *10th IEEE International*

Conference on Fuzzy Systems, pages 924–927, Dec. 2001.
doi:10.1109/FUZZ.2001.1009107.

[36] Joanna C. Hinks, Richard Linares, and John L. Crassidis. Attitude observability from light curve measurements. *AIAA Guidance, Navigation, and Control (GNC) Conference*, Aug. 2013.
doi:10.2514/6.2013-5005.

[37] Jerome Hoepffner, MATTIAS CHEVALIER, THOMAS R. BEWLEY, and DAN S. HENNINGSON. State estimation in wall-bounded flow systems. part 1. perturbed laminar flows. *Journal of Fluid Mechanics*, 534:263–294, 2005.
doi:10.1017/S0022112005004210.

[38] Joshua T. Horwood, Nathan D. Aragon, and Aubrey B. Poore. Gaussian sum filters for space surveillance: Theory and simulations. *Journal of Guidance, Control, and Dynamics*, 34(6):1839–1851, 2012.
doi:10.2514/1.53793.

[39] Joshua T. Horwood and Aubrey B. Poore. Adaptive gaussian sum filters for space surveillance. *IEEE Transactions on Automatic Control*, 56(8):1777–1790, 2011.
doi:10.1109/TAC.2011.2142610.

[40] Yih-Fang Huang, Stefan Werner, Jing Huang, Neelabh Kashyap, and Vijay Gupta. State estimation in electric power grids: Meeting new challenges

presented by the requirements of the future grid. *IEEE Signal Processing Magazine*, 29(5):33–43, 2012.
doi:10.1109/MSP.2012.2187037.

[41] Paul J. Hughes and Kyle T. Alfriend. Covariance based track association with modified equinoctial elements. In *Proceedings of the 2020 AAS/AIAA Astrodynamics Specialist Conference*, 2020. Paper No. AAS 20-681.

[42] Simon J. Illingworth, Jason P. Monty, and Ivan Marusic. Estimating large-scale structures in wall turbulence using linear models. *Journal of Fluid Mechanics*, 842:146–162, 2018.
doi:10.1017/jfm.2018.129.

[43] Bin Jia and Ming Xin. Refined nonlinear gaussian quadrature filter. In *2019 American Control Conference (ACC)*, pages 5366–5371, Jul. 2019.
doi:10.23919/ACC.2019.8814636.

[44] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.
doi:10.2307/2291420.

[45] Simon Julier, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.
doi:10.1109/9.847726.

[46] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the kalman filter to nonlinear systems. *SPIE Proceedings*, 3068, 1997. doi:10.1117/12.280797.

[47] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004. doi:10.1109/JPROC.2003.823141.

[48] John L. Junkins, Maruthi R. Akella, and Kyle T. Alfriend. Non-gaussian error propagation in orbital mechanics. *Journal of the Astronautical Sciences*, 44(4):541–563, October - December 1996.

[49] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, March 1960. doi:10.1115/1.3662552.

[50] Thomas Kelecy, Moriba Jah, and Kyle DeMars. Application of a multiple hypothesis filter to near geo high area-to-mass ratio space objects state estimation. *Acta Astronautica*, 81(2):435–444, 2012. doi:10.1016/j.actaastro.2012.08.006.

[51] Genshiro Kitagawa. Non-gaussian state-space modeling of nonstationary time series (with discussion). *Journal of the American Statistical Association*, 82(400):1032–1063, 1987. doi:10.2307/2289375.

[52] J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2612, Oct 2003. doi:10.1109/TSP.2003.816754.

[53] Jayesh H. Kotecha and Petar M. Djuric. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601, Oct 2003. doi:10.1109/TSP.2003.816758.

[54] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Comment on "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators". *IEEE Transactions on Automatic Control*, 47(8):1406–1408, 2002. doi:10.1109/TAC.2002.800742.

[55] X. Rong Li and Zhanlue Zhao. Measuring estimator's credibility: Noncredibility index. July 2006. doi:10.1109/ICIF.2006.301770.

[56] Xiao-Rong Li and Vesselin P. Jilkov. A survey of maneuvering target tracking: approximation techniques for nonlinear filtering. volume 5428, pages 537–550, August 2004. doi:10.1117/12.553357.

[57] Richard Linares and John L. Crassidis. Space-object shape inversion via adaptive hamiltonian markov chain monte carlo. *Journal of Guidance, Control, and Dynamics*, 41(1):47–58, 2018. doi:10.2514/1.G002296.

[58] Richard Linares, John L. Crassidis, and Moriba K. Jah. Particle filtering light curve based attitude estimation for non-resolved space objects. *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, 152:119–130, Jan. 2014. AAS 14-210.

[59] Richard Linares, John L. Crassidis, and Moriba K. Jah. Space object classification and characterization via multiple model adaptive estimation. In *17th International Conference on Information Fusion (FUSION)*, July 2014.

[60] Richard Linares, Moriba K. Jah, John L. Crassidis, Fred A. Leve, and Tom Kelecy. Astrometric and photometric data fusion for inactive space object mass and area estimation. *Acta Astronautica*, 99:1–15, 2014.
doi:10.1016/j.actaastro.2013.10.018.

[61] Richard Linares, Moriba K. Jah, John L. Crassidis, and Christopher K. Nebelecky. Space object shape characterization and tracking using light curve and angles data. *Journal of Guidance, Control, and Dynamics*, 37(1):13–25, 2014.
doi:10.2514/1.62986.

[62] Richard Linares, Fred A. Leve, Moriba K. Jah, and John L. Crassidis. Space object mass-specific inertia matrix estimation from photometric data. *Advances in the Astronautical Sciences*, 144:41–54, 2012.

[63] Richard Linares, Michael Shoemaker, Andrew Walker, Piyush M. Mehta, David M. Palmer, David C. Thompsonk, Josef Koller, and John L. Crassidis.

Photometric data from non-resolved objects for space object characterization and improved atmospheric modeling. *Advanced Maui Optical and Space Surveillance Technologies Conference*, 1:32, 2013.

[64] Bo Liu, Boujemaa Ait-El-Fquih, and Ibrahim Hoteit. Efficient kernel-based ensemble gaussian mixture filtering. *Monthly Weather Review*, 144(2):781–800, 2016.
doi:10.1175/MWR-D-14-00292.1.

[65] Vishnuu Mallik and Moriba K. Jah. Reconciling space object observed and solar pressure albedo-areas via astrometric and photometric data fusion. *Advances in Space Research*, 63(1):404–416, 2019.
doi:10.1016/j.asr.2018.08.005.

[66] F. Landis Markley and John L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Space Technology Library. Springer, 2014.

[67] Peter S. Maybeck. *Stochastic Models, Estimation, and Control, Volume 2*. Academic Press, New York, NY, 1982.

[68] Jay W. McMahon and Daniel J. Scheeres. Improving space object catalog maintenance through advances in solar radiation pressure modeling. *Journal of Guidance, Control, and Dynamics*, 38(8):1366–1380, 2015.
doi:10.2514/1.G000666.

[69] A. Monticelli. Electric power system state estimation. *Proceedings of the IEEE*, 88(2):262–282, 2000.
doi:10.1109/5.824004.

[70] Masaya Murata, Isao Kawano, and Koichi Inoue. Extended, unscented kalman, and sigma point multiple distribution estimation filters for nonlinear discrete state-space models. *IEEE Control Systems Letters*, 4(4):982–987, 2020.
doi:10.1109/LCSYS.2020.2998578.

[71] National Research Council. *Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards*. The National Academies Press, 2012.

[72] Ionel M. Navon. Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dynamics of Atmospheres and Oceans*, 27(1-4):55–79, 1998.
doi:10.1016/S0377-0265(97)00032-8.

[73] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *21st International Conference on Machine Learning*, Jul. 2004.
doi:10.1145/1015330.1015435.

[74] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D.J. Patil, and James A. Yorke.

A local ensemble kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):415–428, 2004. doi:10.3402/tellusa.v56i5.14462.

[75] F. Papi, M. Podt, Y. Boers, G. Battistello, and M. Ulmke. On constraints exploitation for particle filtering based target tracking. *15th International Conference on Information Fusion*, pages 455–462, Aug. 2012.

[76] Byeong U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990. doi:10.1080/01621459.1990.10475307.

[77] E. Pate-Cornell and M. Sachon. Risks of particle hits during space walks in low earth orbit. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):134–146, 2001. doi:10.1109/7.913673.

[78] Nikolaos K. Pavlis, Simon A. Holmes, Steve C. Kenyon, and John K. Factor. An Earth gravitational model to degree 2160: EGM2008. In *Proceedings of the European Geosciences Union General Assembly*, Vienna, Austria, April 13-18 2008.

[79] Michael K. Pitt and Neil Shephard. Filtering via simulation: auxiliary particle. *Journal of the American Statistical Association*, 446(94):590–599, 1999. doi:10.1049/ip-f-2.1993.0015.

[80] Mark L Psiaki. The blind tricyclist problem and a comparative study of nonlinear filters. *IEEE Control Systems Magazine*, 33(3):40–54, 2013. doi:10.1109/MCS.2013.2249422.

[81] Mark L Psiaki. Gaussian mixture nonlinear filtering with resampling for mixand narrowing. *IEEE Transactions on Signal Processing*, 64(21):5499–5512, 2016. doi:10.1109/TSP.2016.2595503.

[82] Mark L Psiaki, Jonathan R Schoenberg, and Isaac T Miller. Gaussian sum reapproximation for use in a nonlinear filter. *Journal of Guidance, Control, and Dynamics*, 38(2):292–303, 2015. doi:10.2514/1.G000541.

[83] D. Raihan and S. Chakravorty. Particle gaussian mixture (pgm) filters. pages 1369–1376, July 2016. ISBN:978-0-9964-5274-8.

[84] Dilshad Raihan and Suman Chakravorty. Particle gaussian mixture filters-i. *Automatica*, 98:331–340, 2018. doi:10.1016/j.automatica.2018.07.023.

[85] Matti Raitoharju and Simo Ali-Loytty. An adaptive derivative free method for bayesian posterior approximation. *IEEE Signal Processing Letters*, 19(2):87–90, 2012. doi:10.1109/LSP.2011.2179800.

[86] Ilia Rapoport and Itzhack Y. Bar-Itzhack. On the information dilution theorem and its application to attitude determination. *The Journal of the Astronautical Sciences*, 49(3):489–508, 2001.

[87] Matthew Richardson, Tom Kelecy, Jason Stauch, Charles J. Wetterer, and Channing Chow. Mass estimation through fusion of astrometric and photometric data with applications to orbital debris characterization. *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, 168:2209–2227, Jan. 2019. AAS 19-268.

[88] Chris Sabol, Thomas Sukut, Keric Hill, Kyle T. Alfriend, Brendan Wright, You Li, and Paul W. Schumacher, Jr. Linearized orbit covariance generation and propagation analysis via simple monte carlo simulations. In *20th Annual AAS/AIAA Space Flight Mechanics Meeting*, San Diego, California, February 15-17 2010.

[89] Miyamoto Sadaaki and Mukaidono Masao. Fuzzy c-means as a regularization and maximum entropy approach. In *7th International Fuzzy Systems Association World Congress*, volume 2, pages 86–92, Jun. 1997.

[90] Simo Sarkka. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
doi:10.1017/CBO9781139344203.

[91] S. F. Schmidt. Application of state-space methods to navigation problems. *Advances in Control Systems*, 3:293–340, 1966.
doi:10.1016/B978-1-4831-6716-9.50011-4.

[92] Oliver C Schrempf, Dietrich Brunn, and Uwe D Hanebeck. Density approximation based on dirac mixtures with regard to nonlinear estimation and filtering. In *Proceedings of the 2006 IEEE Conference on Decision and Control (CDC 2006)*, San Diego, California, 2006. doi:10.1109/CDC.2006.376759.

[93] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons, New Jersey, NJ, 1992. doi:10.1002/0470045345.

[94] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[95] Dan Simon. *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley and Sons, New Jersey, NJ, 2006. pp. 212-223, doi:10.1002/0470045345.

[96] Dan Simon and Donald L. Simon. Constrained kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, 2010. doi:10.1080/00207720903042970.

[97] H. W. Sorenson and D. L. Alspach. Recursive Bayesian Estimation Using Gaussian Sums. *Automatica*, 7(4):465–479, July 1971. doi:10.1016/0005-1098(71)90097-5.

[98] Byron D. Tapley, Bob E. Schutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[99] Bruno O. S. Teixeira, Leonardo A. B. Tôrres, Luis A. Aguirre, and Dennis S. Bernstein. On unscented kalman filtering with state interval constraints. *Journal of Process Control*, 20(1):45–57, 2010. doi:10.1016/j.jprocont.2009.10.007.

[100] Bruno O.S. Teixeira, Mario A. Santillo, R. Scott Erwin, and Dennis S. Bernstein. Spacecraft tracking using sampled-data kalman filters. *IEEE Control Systems Magazine*, 28(4):78–94, 2008. doi:10.1109/MCS.2008.923231.

[101] Gabriel Terejanu, Puneet Singla, Tarunraj Singh, and Peter D. Scott. Adaptive gaussian sum filter for nonlinear bayesian estimation. *IEEE Transactions on Automatic Control*, 56(9):2151–2156, 2011. doi:10.1109/TAC.2011.2141550.

[102] Petr Tichavsky, Carlos H. Muravchik, and Arye Nehorai. Posterior cramer-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998. doi:10.1109/78.668800.

[103] Kirsten Tuggle and Renato Zanetti. Automated Splitting Gaussian Mixture Nonlinear Measurement Update. *Journal of Guidance, Control, and Dynamics*, 41(3):725–734, March 2018. doi:10.2514/1.G003109.

[104] Pramod Vachhani, Shankar Narasimhan, and Raghunathan Rengaswamy. Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation. *Journal of Process Control*, 16(10):1075–1086, 2006. doi:10.1016/j.jprocont.2006.07.002.

[105] David A. Vallado. *Fundamentals of Astrodynamics and Applications, 3rd Edition*. McGraw-Hill, New Jersey, NJ, 1997.

[106] Vivek Vittaldev and Ryan P Russell. Collision probability for space objects using gaussian mixture models. In *Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Meeting*, volume 148, pages 2339–2358. Advances in the Astronautical Sciences, Univelt San Diego, CA, 2013. ISBN:978-0-8770-3597-8.

[107] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000. doi:10.1109/ASSPCC.2000.882463.

[108] Charles J. Wetterer, C. Channing Chow, John L. Crassidis, Richard Linares, and Moriba K. Jah. Simultaneous position, velocity, attitude, angular rates, and surface parameter estimation using astrometric and photometric observations. *Proceedings of the 16th International Conference on Information Fusion*, pages 997–1004, July 2013.

[109] Charles J. Wetterer, Bobby Hunt, Kris Hamada, John L. Crassidis, and Paul Kervin. Shape, surface parameter, and attitude profile estimation using a

multiple hypothesis unscented kalman filter. *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, 152:1475–1484, Jan. 2014. AAS 14-303.

[110] Charles J. Wetterer and Moriba K. Jah. Attitude estimation from light curves. *Journal of Guidance, Control, and Dynamics*, 32(5):1648–1651, 2009.
doi:10.2514/1.44254.

[111] Drew P. Woodbury and John L. Junkins. On the consider kalman filter. *AIAA Guidance, Navigation, and Control Conference*, Aug. 2010. Paper No. AIAA 2010-7752.

[112] Sehyun Yun and Renato Zanetti. Sequential monte carlo filtering with gaussian mixture sampling. *Journal of Guidance, Control, and Dynamics*, 42(9):2069–2077, 2019.
doi:10.2514/1.G004403.

[113] Sehyun Yun and Renato Zanetti. Nonlinear filtering of light-curve data. *Advances in Space Research*, 66(7):1672–1688, 2020.
doi:10.1016/j.asr.2020.06.024.

[114] Renato Zanetti. Recursive update filtering for nonlinear estimation. *IEEE Transactions on Automatic Control*, 57(6):1481–1490, 2012.
doi:10.1109/TAC.2011.2178334.

[115] Renato Zanetti. Rotations, transformations, left quaternions, right quaternions? *The Journal of the Astronautical Sciences*, 66(3):361–381, 2019.

doi:10.1007/s40295-018-00151-2.

[116] Renato Zanetti, Kyle J. DeMars, and Robert H. Bishop. Underweighting nonlinear measurements. *Journal of Guidance, Control, and Dynamics*, 33(5):1670–1675, 2010.
doi:10.2514/1.50596.

[117] Renato Zanetti and Kirsten Tuggle. A novel gaussian mixture approximation for nonlinear estimation. In *21st International Conference on Information Fusion (FUSION)*, Jul. 2018.
doi:10.23919/ICIF.2018.8455485.